# SpeedStream™ Router Family

## Command Line Interface Guide

January 2000

## Copyright

Efficient Networks provides this publication “as is” without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

All rights reserved. No part of this book may be reproduced in any form or by any means without written permission from Efficient Networks.

Changes are periodically made to the information in this book. They will be incorporated in subsequent editions. Efficient Networks may make improvements and/or changes in the product described in this publication at any time.

© Copyright 1996-2000 Efficient Networks, Inc.

## Trademarks

Efficient Networks® is a registered trademark of Efficient Networks, Inc.

SpeedStream is a trademark of Efficient Networks, Inc.

All other trademarks and registered trademarks mentioned in this manual are the sole property of their respective companies.

# What's New in Release 4?

This version of the Command Line Interface (CLI) manual has been updated to document features available with Release 4 of the kernel software. The following list directs you to the CLI documentation for these new features:

**See page 18 to read about:**

**Telephony Services**

- A Voice over DSL (VoDSL) router allows delivery of both telephony (voice) and data services over a single DSL line.

**See page 76 to read about:**

**Multiple IP Subnets**

- Each Ethernet port can provide service to more than one IP subnetwork.
- Each IP subnet is referenced by its own logical Ethernet interface.
- This feature supports NAT, RIP, DHCP, ARP, and IP filters.

**See page 77 to read about:**

**Virtual Routing Tables**

- Each table routes a range of source IP addresses.

**See page 119 to read about:**

**IPSec** (Internet Protocol Security), including:

- **Internet Key Exchange** (IKE)
- Main and aggressive IKE modes
- Tunnel and transport encapsulation modes
- AH and ESP security protocols
- DES and 3DES encryption

**See page 223 to read about:**

**SDSL Nokia EOC**

- Embedded operations channel allows CO to set line speed.
- Release 5 of Nokia SDSL DSLAM support

**See page 172 and page 191 to read about:**

**IP Filter Enhancements**

- Line numbers to manage filter list order

- Clear command to reset filter counters.
- Watch message control via -q (quiet) and -v (verbose) parameters.

**See page 99 to read about:**

**Software Option Keys**

- New command to enter a software option key

**See page 176 and page 203 to read about:**

**Setting a Management Address**

- Assignment of an IP address for management use only

**See page 293 to read about:**

**Debugging Commands**

- Debugging commands should be used with caution because they are not fully supported.

# About This Guide

The *Command Line Interface* guide contains information on the syntax and use of the Command Line Interface for the family of DSL routers. It provides the steps and information needed to configure the router software and troubleshoot problems using the Command Line Interface. Configuration of network connections, bridging, routing, and security features are essentially the same for all DSL routers, unless otherwise noted. The guide also provides detailed information about the system's bridging, routing, addressing, and security operations.

This guide is intended for small and home office users, remote office users, and other networking professionals who are installing and maintaining bridged and routed networks.

## How This Guide is Organized

This guide is intended to help you configure and manage the router using the Command Line Interface. The guide assumes that you have read the information about the router and installed the hardware using the *Internet Quick Start Guide*. The guide is divided into eight parts:

**Introduction.** Describes the features of the Command Line Interface.

**Advanced Topics.** Contains additional information on topics such as interoperability, routing and bridging operations, PAP/CHAP security negotiation, bandwidth management, protocol conformance, and the file system.

**Planning for Router Configuration.** Provides information unique to configuration using the Command Line Interface including worksheets for collecting required information.

**Configuring Router Software.** Describes how to configure the router using the Command Line Interface.

**Configuring Special Features.** Describes how to configure features such as Bridging Filtering, RIP, DHCP, NAT, Management Security, Software Options Keys, Encryption, IP Filtering, and L2TP Tunneling.

**Command Line Interface Reference.** Describes the syntax of each command and the results when the command is entered.

**Managing the Router.** Describes SNMP management capabilities, TFTP client and server, TELNET support and how to upgrade the system software, boot code, backup and restore configuration files, FLASH memory recovery procedures, and batch file command execution.

**Troubleshooting.** Describes diagnostic tools used for identifying and correcting hardware and software problems.

# References

*User Guide*. Contains an overview of the router's software and hardware features and details on hardware installation and software configuration using the Windows-based Configuration Manager.

*Quick Start Guide*. Describes the configuration process involved in setting up a specific router model.

# Typographic Conventions

The following conventions are used in this guide:

| Item | Type Face | Examples |
|---|---|---|
| Book titles, command reference parameters, reference to a specific section/chapter in this guide, emphasis in text. | Italics | Refer to Chapter 1. *Advanced Features*<br><br>system name <*name*> |
| Keywords in command reference instructions | Bold | **save** |
| Examples showing you what to type and what is displayed on the terminal. | Mono-spaced font | `remote listIpRoute hq` |
| File names | Upper case | Copy file CFGMGR.EXE |

# Table of Contents


What's New in Release 4? . . . . . 3

About This Guide. . . . . 5

How This Guide is Organized . . . . . 5

References . . . . . 6

Typographic Conventions . . . . . 6

Table of Contents . . . . . 7

Introduction . . . . . 13

**Chapter 1. Router Concepts . . . . . 15**

Routing and Bridging. . . . . 15

Routing . . . . . 15

Bridging. . . . . 16

When to Use Routing or Bridging or Both . . . . . 16

How Routing and Bridging Work Together . . . . . 17

Routing and Bridging Controls . . . . . 17

Bridge Filtering . . . . . 18

Telephony Services . . . . . 18

Configuring Your Telephony Services . . . . . 19

PAP/CHAP Security Authentication . . . . . 20

Authentication Process . . . . . 21

Security Passwords and Levels . . . . . 21

Interoperability Between the Router and Other Equipment . . . . . 23

Protocol Conformance . . . . . 23

IP Routing . . . . . 24

IPX Routing . . . . . 24

Encapsulation Options . . . . . 24

PPP . . . . . 24

PPPLLC. . . . . 25

RFC 1483 or RFC 1490 . . . . . 25

MAC Encapsulated Routing: RFC 1483MER (ATM) or RFC 1490MER (Frame Relay) . . . . . 26

FRF8 . . . . . 26

rawIP . . . . . 26

System Files . . . . . 26

**Chapter 2. Planning for Router Configuration. . . . . 29**

Important Terminology . . . . . 29

Essential Configuration Information . . . . . 30

PPP Link Protocol (over ATM or Frame Relay). . . . . 31

RFC 1483/RFC 1490 Link Protocols . . . . . 36

MAC Encapsulated Routing . . . . . 39

FRF8 Link Protocol. . . . . 41

Dual-Ethernet Router Configuration. . . . . 42

**Chapter 3. Configuring Router Software . . . . . 44**

Configuration Tables . . . . . 45

Configuring PPP with IP Routing . . . . . 46

Configuring PPP with IPX Routing . . . . . 47

Configuring PPP with Bridging . . . . . 48

Configuring PPP over Ethernet (PPPoE) . . . . . 49

Configuring RFC 1483 / RFC 1490 with IP Routing . . . . . 50
Configuring RFC 1483 / RFC 1490 with IPX Routing . . . . . 51
Configuring RFC 1483 / RFC 1490 with Bridging . . . . . 52
Configuring MAC Encapsulated Routing: RFC 1483MER / RFC 1490MER with IP Routing . 53
Configuring FRF8 with IP Routing . . . . . 54
Configuring Mixed Network Protocols . . . . . 55
Configuring a Dual-Ethernet Router for IP Routing . . . . . 56
Verify the Router Configuration . . . . . 57
Test IP Routing . . . . . 57
Test Bridging to a Remote Destination. . . . . 57
Test IPX Routing . . . . . 58
Sample Configurations . . . . . 59
Sample Configuration 1: PPP with IP and IPX . . . . . 59
Sample Configuration 2: RFC 1483 with IP and Bridging. . . . . 67
Sample Configuration 3: Configuring a Dual-Ethernet Router for IP Routing . . . . . 73
**Chapter 4. Configuring Special Features . . . . . 75**
Multiple IP Subnets . . . . . 76
Virtual Routing Tables . . . . . 77
Bridge Filtering and IP Firewall . . . . . 78
Configure Bridge Filtering . . . . . 78
Enable/Disable Internet Firewall Filtering . . . . . 79
IP (RIP) Protocol Controls . . . . . 80
DHCP (Dynamic Host Configuration Protocol) . . . . . 81
DHCP Address Allocation . . . . . 81
DHCP Client Requests . . . . . 81
DHCP Administration and Configuration. . . . . 82
Manipulating Subnetworks and Explicit Client Leases . . . . . 82
Setting Option Values . . . . . 85
Managing BootP . . . . . 87
Defining Option Types . . . . . 88
Configuring BootP/DHCP Relays . . . . . 89
DHCP Information File . . . . . 90
Network Address Translation (NAT) . . . . . 91
General NAT Rules . . . . . 91
Masquerading . . . . . 91
Classic NAT . . . . . 95
Client Configuration . . . . . 95
Controlling Remote Management . . . . . 97
Disabling Remote Management . . . . . 97
Re-enabling Remote Management . . . . . 97
Validating Clients . . . . . 97
Restricting Remote Access . . . . . 98
Changing the SNMP Community Name . . . . . 98
Disabling WAN Management . . . . . 98
Software Option Keys . . . . . 99
Listing the Installed Software Options . . . . . 99
Adding a New Software Option Key . . . . . 99
Encryption . . . . . 100
PPP DES (RFC 1969) Encryption . . . . . 100
Diffie-Hellman Encryption . . . . . 101
IP Filtering . . . . . 103
Filters and Interfaces . . . . . 103
Configuring Filters with Network Address Translation Enabled . . . . . 104

Filter Actions . . . . . 104
IP Filter Commands . . . . . 105
Special Notes . . . . . 105
L2TP Tunneling — Virtual Dial-Up . . . . . 106
Introduction . . . . . 106
L2TP Concepts . . . . . 106
Configuration . . . . . 109
Sample Configurations . . . . . 111
IPSec (Internet Protocol Security) . . . . . 119
Transport and Tunnel Encapsulation Modes . . . . . 119
ESP and AH Security Protocols . . . . . 120
IKE Management . . . . . 121
Main Mode and Aggressive Mode . . . . . 122
Additional IKE Settings . . . . . 122
Security Associations (SAs) . . . . . 123
IKE Commands . . . . . 123
IKE Peer Commands . . . . . 124
IKE Proposal Commands . . . . . 125
IKE IPSec Proposal Commands . . . . . 125
IKE IPSec Policy Commands . . . . . 127
IKE Configuration Examples . . . . . 128
Main Mode Example . . . . . 128
Aggressive Mode Example . . . . . 131
IPSec Commands . . . . . 132

**Chapter 5. Command Line Interface Reference . . . . . 135**

Command Line Interface Conventions . . . . . 135
System-Level Commands . . . . . 136
File System Commands . . . . . 146
Frame Commands . . . . . 149
Router Configuration Commands . . . . . 152
Target Router System Configuration Commands (SYSTEM) . . . . . 153
Target Router Ethernet LAN Bridging and Routing (ETH) . . . . . 167
Remote Router Access Configuration (REMOTE) . . . . . 182
Asymmetric Digital Subscriber Line Commands (ADSL) . . . . . 208
Asynchronous Transfer Mode Commands (ATM) . . . . . 210
DMT Commands . . . . . 212
Dual-Ethernet Router Commands (ETH) . . . . . 213
High-Speed Digital Subscriber Line Commands (HDSL) . . . . . 217
ISDN Digital Subscriber Line (IDSL) . . . . . 220
SDSL Commands . . . . . 222
DHCP (Dynamic Host Configuration Protocol) Commands . . . . . 227
L2TP — Virtual Dial-Up Configuration (L2TP) . . . . . 238
Bridge Filtering Commands (FILTER BR) . . . . . 246
Internet Key Exchange (IKE) Commands . . . . . 248
IPSec Commands . . . . . 261

**Chapter 6. Managing the Router . . . . . 267**

Simple Network Management Protocol (SNMP) . . . . . 267
Telnet Remote Access . . . . . 268
Client TFTP Facility . . . . . 268
TFTP Server . . . . . 268
BootP Server . . . . . 269
Boot Code . . . . . 269

Manual Boot Menu . . . . . 269
Identifying Fatal Boot Failures . . . . . 273
Software Kernel Upgrades . . . . . 273
Booting and Upgrading from the LAN . . . . . 273
Upgrading from the WAN Line . . . . . 275
Backup and Restore Configuration Files . . . . . 276
Backup Configuration Files (Recommended Procedure) . . . . . 276
Restore Configuration Files . . . . . 276
FLASH Memory Recovery Procedures . . . . . 277
Recovering Kernels for Routers with Configuration Switches . . . . . 277
Recovering Kernels for Routers with a Reset Button . . . . . 278
Recovering Passwords and IP Addresses . . . . . 279
Routers with Configuration Switches . . . . . 279
Routers with a Reset Button . . . . . 279
Batch File Command Execution . . . . . 279

**Chapter 7. Troubleshooting . . . . . 281**
Diagnostic Tools . . . . . 281
Using LEDs . . . . . 281
History Log . . . . . 282
Ping Command . . . . . 283
Investigating Hardware Installation Problems . . . . . 284
Investigating Software Configuration Problems . . . . . 285
Problems Connecting to the Router . . . . . 285
Problems with the Login Password . . . . . 285
Problems Accessing the Remote Network . . . . . 286
Problems Accessing the Router via Telnet . . . . . 288
Problems Downloading Software . . . . . 288
Trouble-Shooting Telephony Services . . . . . 288
System Messages . . . . . 290
Time-Stamped Messages . . . . . 290
How to Obtain Technical Support . . . . . 292
Debugging Commands . . . . . 293
General Debug Commands . . . . . 293
ATM Debug Commands . . . . . 294
Web GUI Debug Commands . . . . . 294
SDSL Debug Commands . . . . . 295
Voice Router Debug Commands . . . . . 296
ADSL DMT Router Debug Commands . . . . . 297
Frame Relay Debug Commands . . . . . 297
ATM Tracing Commands . . . . . 297

**Appendix A. Network Information Worksheets . . . . . 299**
Configuring PPP with IP Routing . . . . . 300
Configuring PPP with IPX Routing . . . . . 301
Configuring PPP with Bridging . . . . . 302
Configuring RFC 1483 / RFC 1490 with IP Routing . . . . . 303
Configuring RFC 1483 / RFC 1490 with IPX Routing . . . . . 304
Configuring RFC 1483 / RFC 1490 with Bridging . . . . . 305
Configuring RFC 1483MER / RFC 1490MER with IP Routing . . . . . 306
Configuring FRF8 with IP Routing . . . . . 307
Configuring a Dual-Ethernet Router for IP Routing . . . . . 308

**Appendix B. Configuring IPX Routing . . . . . 309**

IPX Routing Concepts . . . . . 309
Configure IPX Routing . . . . . 309
Step 1: Collect Your Network Information for the Target (Local) Router . . . . . 310
Step 2: Review your Settings . . . . . 311

**Appendix C. Accessing the Command Line Interface . . . . . 313**
Connect the PC to the Console Port of the Router . . . . . 313
Access the Command Line Interface . . . . . 313
Terminal Window under Configuration Manager . . . . . 313
Terminal Session under Windows (HyperTerminal) . . . . . 314
Terminal Session for a Non-Windows Platform (Macintosh or UNIX) . . . . . 314
Telnet Session . . . . . 314

**Index . . . . . 315**

# Introduction

This manual describes the Command Line Interface for your router. The Command Line Interface gives you access to all the capabilities of your router.

Many of the router configuration capabilities are also available through an easy-to-use, point-and-click graphic interface.Your router supports use of the Microsoft® Windows™-based Configuration Manager, Quick Start, and/or Web-based GUI programs. To learn how to access those programs, see the documentation that came with the router.

To learn how to access the Command Line Interface from Configuration Manager or Quick Start, see *Accessing the Command Line Interface,* page 313.

Using the Command Line Interface allows you to perform the following basic configuration tasks:

- Set names, passwords, PVC numbers, and link and network parameters
- Configure specific protocol requirements, such as IP or IPX addresses and IP protocol controls
- Activate bridging and routing protocols
- Enable the Internet firewall filter with IP routing

The Command Line Interface also allows you to perform the following advanced tasks:

- Manage the router's file system
- Set bridging filters
- Configure the type of DSL technology specific to your router (e.g., ADSL, SDSL)
- Configure the Dual-Ethernet router
- Issue online status commands
- Monitor error messages
- Set RIP options
- Create and execute script files
- Configure advanced features, including:
  - DHCP
  - NAT
  - Telnet/SNMP security
  - Host mapping
  - IP multicast
  - Encryption
  - IP filtering
  - L2TP tunneling
  - IP security

# Chapter 1. Router Concepts

This chapter provides background information applicable to the router on topics useful to network administrators. These topics include:

- Routing and bridging
  - Routing and bridging controls
  - Bridge filtering
- Voice routing
- PAP/CHAP security authentication
- Security passwords and levels
- Interoperability between the router and other systems
- Protocol conformance
- Encapsulation options
- System files

## Routing and Bridging

The router can operate as a bridge, a router, or as both (sometimes called a brouter). The following sections describe routing and bridging and how the two functions operate together.

### Routing

Routing is the process that determines where data is sent. A router can route user data from source to destination over different LAN and WAN links. Routing relies on routing address tables to determine the best path for each packet to take.

The routes within a routing address table are established in two ways:

- The routing tables can be seeded, that is, addresses for remote destinations are placed in the table along with path details and the associated costs (path latency).
- The routing tables can also be built dynamically; i.e., the location of remote stations, hosts, and networks are updated from broadcast packet information.

Routing helps to increase network capacity by localizing traffic on LAN segments. It also provides security by isolating traffic on segmented LANs. Routing extends the reach of networks beyond the limits of each LAN segment.

Numerous network protocols have evolved, and within each protocol are associated protocols for routing, error handling, network management, etc. The following chart displays the networking and associated protocols supported by the router.

| Network Protocol | Associated Protocols | Description |
|---|---|---|
| **Internet Protocol (IP)** | Routing Information Protocol (RIP) | Maintains a map of the network |
| | Address-Resolution Protocol (ARP) | Maps IP addresses to datalink addresses |
| | Reverse Address Resolution Protocol (RARP)[a] | Maps data-link addresses to IP addresses |
| | Internetwork Control Message Protocol (ICMP) | Diagnostic and error reporting/ recovery |
| | Simple Network Management Protocol (SNMP) | Network management |
| **Internet Packet Exchange (IPX)** | Routing Information Protocol (RIP)[b] | Maintains a map of the network |
| | Service Advertising Protocol (SAP) | Distributes information about service names and addresses |

a Used only during a network boot.
b IPX-RIP is a different protocol from IP-RIP and it includes time delays.

## Bridging

Bridging connects two or more LANs so that all devices share the same logical LAN segment and network number. The MAC layer header contains source and destination addresses used to transfer frames.
An address table is dynamically built and updated with the location of devices when the frames are received.

Transparent bridging allows locally connected devices to send frames to all devices as if they were local.

Bridging allows frames to be sent to all destinations regardless of the network protocols used. It allows protocols that cannot be routed (such as NETBIOS) to be forwarded and allows optimizing internetwork capacity by localizing traffic on LAN segments. A bridge extends the physical reach of networks beyond the limits of each LAN segment. Bridging can increase network security with filtering.

The router bridging support includes the IEEE 802.1D standard for LAN-to-LAN bridging and the Spanning Tree Protocol for interoperability with other vendors' bridge/routers. Bridging is provided over PPP as well as adjacent LAN ports.

## When to Use Routing or Bridging or Both

The following charts describe the operational characteristics of the router when you enable routing, bridging, or both routing and bridging.

| IP/IPX Routing On | Bridging to/from Remote Router Off |
|---|---|
| Data packets carried | IP (TCP, UDP), IPX |
| Operational characteristics | Basic IP, IPX connectivity |
| Typical usage | When only IP/IPX traffic is to be routed and all other traffic is to be ignored. For IP, used for Internet access.<br>**Note:** This is the most easily controlled configuration. |

| IP/IPX Routing On | Bridging to/from Remote Router On |
|---|---|
| Data packets carried | IP/IPX routed; all other packets bridged. |
| Operational characteristics | IP/IPX routing and allows other protocols, such as NetBEUI (that can't be routed), to be bridged. |
| Typical usage | When only IP/IPX traffic is to be routed but some non-routed protocol is required. Used for client/server configurations. |

| IP/IPX Routing Off | Bridging to/from Remote Router On |
|---|---|
| Data packets carried | All packets bridged. |
| Operational characteristics | Allows protocols, such as NetBEUI (that can't be routed) to be bridged. |
| Typical usage | Peer-to-peer bridging and when the remote end supports only bridging. |

## How Routing and Bridging Work Together

The router follows these rules when operating as both a router and a bridge:

- The router operates as a router for network protocols that are enabled for routing (IP or IPX).
- The router operates as a bridge for protocols that are not supported for routing.
- Routing takes precedence over bridging; i.e., when routing is active, the router uses the packet's protocol address information to route the packet.
- If the protocol is not supported, the router will use the MAC address information to forward the packet.

## Routing and Bridging Controls

The router can be configured to perform general routing and bridging while allowing you to set specific controls.

- One remote router can be designated as the outbound default bridging destination. All outbound bridging traffic with an unknown destination is sent to the default bridging destination.
- Bridging from specific remote routers can be controlled by enabling or disabling bridging from individual remote routers.

- Routing is performed to all remote routers entered into the remote router database. All routing can be enabled or disabled with a system-wide control.

Operation of the router is influenced by routing and bridging controls and filters set during router configuration as well as automatic spoofing and filtering performed by the router. For example, general IP or IPX routing, and routing or bridging from specific remote routers are controls set during the configuration process.

Spoofing and filtering, which minimize the number of packets that flow across the WAN, are performed automatically by the router. For example, RIP routing packets and certain NetBEUI packets are spoofed even if only bridging is enabled.

## Bridge Filtering

You can control the flow of packets across the router using bridge filtering. Bridge filtering lets you “deny”or “allow” packets to cross the network based on position and hexadecimal content within the packet. This feature lets you restrict or forward messages with a specified address, protocol, or data content. Common uses are to prevent access to remote networks, control unauthorized access to the local network, and limit unnecessary traffic.

For example, it might be necessary to restrict remote access for specific users on the local network. In this case, bridging filters are defined using the local MAC address for each user to be restricted. Each bridging filter is specified as a “deny” filter based on the MAC address and position of the address within the packet. Deny filtering mode is then enabled to initiate bridge filtering. No packet with one of the MAC addresses can be bridged across the router until the deny filtering mode is disabled.

Similarly, protocol filtering can be used to prevent a specific protocol from being bridged. In this case, the protocol ID field in a packet is used to deny or allow a packet. You can also restrict, for example, the bridging of specific broadcast packets.

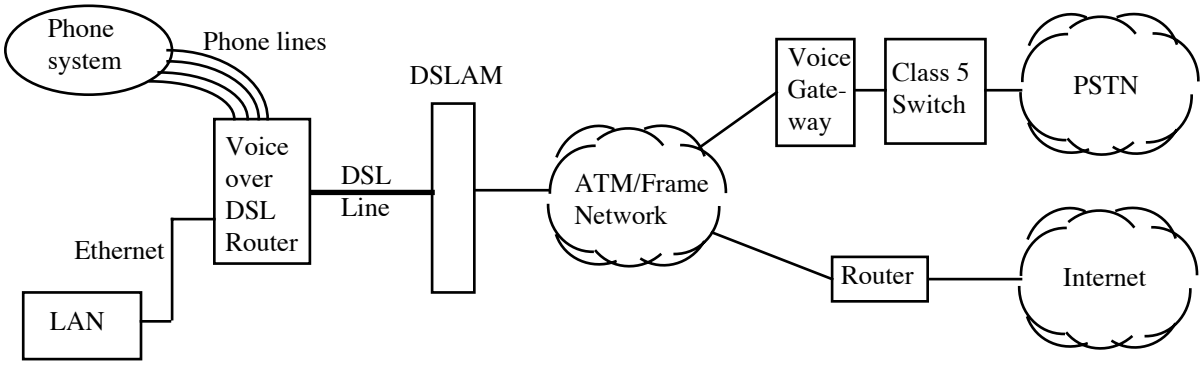
# Telephony Services

A Voice over DSL (VoDSL) router allows the delivery of both telephony (voice) and data services over a single DSL line. It acts as an Integrated Access Device (IAD), residing on the customer premises and connecting to a DSL circuit. As such, it serves as a circuit/packet gateway and provides standard telephone service as well as Internet service via an Ethernet connection. Thus, the user has access to toll-quality telephone lines and continuous, high-speed Internet and remote LAN services over a single copper loop.

Features provided in your VoDSL router may include:

- Support for Jetstream and CopperCom gateways
- Upstream traffic shaping (bandwidth management) of data when the telephony interface is active
- ADPCM or PCM voice encoding
- Local echo canceling (G.168)

This diagram illustrates how a Voice over DSL router connects both a phone system to the PSTN and a LAN to the Internet over the same DSL line.




## Configuring Your Telephony Services

Router models are available to support telephony services over both ATM and Frame Relay networks. For telephony over ATM, the VPI/VCI is automatically set (to 0*39 for most routers). For telephony over Frame Relay, the DLCI is automatically set to 22. The value must match your service provider's value. You can change the value if necessary (see Troubleshooting Telephony Services, page 288).

The phone number for each port is set by your voice service provider.

The phone dial tone is provided by the Class 5 switch via the voice gateway at your regional switching center (RSC). All voice features of the switch are passed through to the phone set. The router supports the calling services that you subscribe to from your service provider, such as call forwarding, caller ID, messaging, etc.

The bandwidth required for a voice call depends on whether PCM or ADPCM encoding is used. With PCM, each voice call takes about 80 Kb of bandwidth (64 Kb plus overhead) when the phone goes off hook. With ADPCM, only about 40 Kb is used (32 Kb plus overhead).

Configuration for voice and data routing can be performed using the Web-based Easy Setup configuration program.

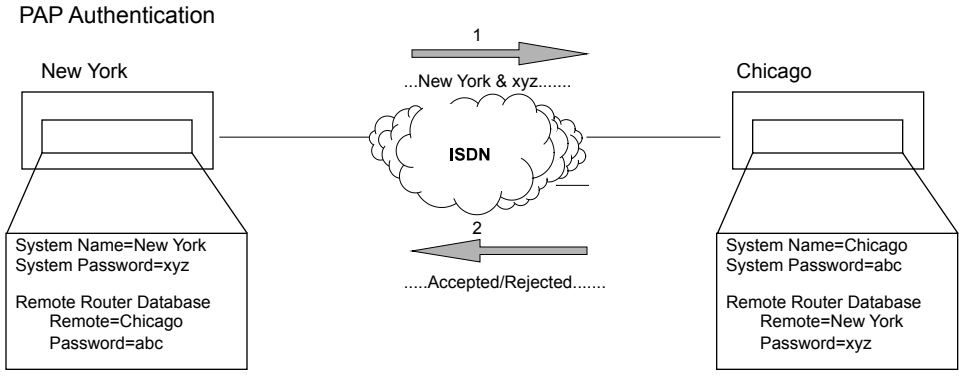
You can use the Port Monitor GUI program to see the voice PVC and the last event message. To see LMI statistics for a frame relay router, use the **frame stats** command (page 150). To see AAL2 statistics for the voice gateway, use the **voice l2stats** command (Jetstream gateway only). For other commands to monitor telephony services, see Trouble-Shooting Telephony Services (page 288).

# PAP/CHAP Security Authentication

The router supports PAP (Password Authentication Protocol) and CHAP (Challenge Handshake Authentication Protocol) under PPP.

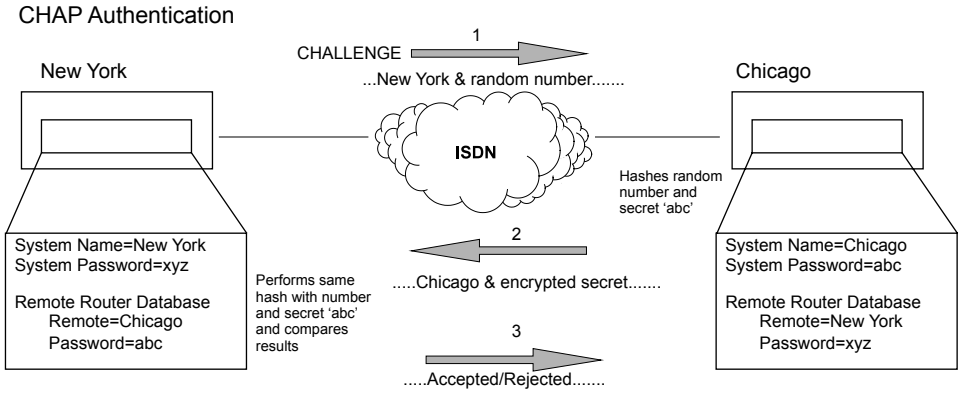
Security authentication may not be required due to the nature of the connection in a DSL environment (traffic occurs on a dedicated line/virtual circuit. However, authentication may be specifically required by the remote end, the ISP, or the NSP. When authentication is not required, security can be disabled with the command **remote disauthen** (page 190).

PAP provides verification of passwords between routers using a two-way handshake. One router (peer) sends the system name and password to the other router. Then the other router (known as the authenticator) checks the peer's password against the configured remote router's password and returns acknowledgment.




CHAP is more secure than PAP because unencrypted passwords are not sent across the network. CHAP uses a three-way handshake. One router (known as the authenticator) challenges the other router (known as the peer) by generating a random number and sending it along with the system name. The peer then applies a one-way hash algorithm to the random number and returns this encrypted information along with the system name.

The authenticator then runs the same algorithm and compares the result with the expected value. This authentication method depends upon a password or secret known only to both ends.



## Authentication Process

The authentication process occurs regardless of whether a remote router connects to the local router or vice versa, and even if the remote end does not request authentication. It is a *bi-directional process*, where each end can authenticate the other using the protocol of its choice (provided the other end supports it).

During link negotiation (LCP), each side of the link negotiates which protocol to use for authentication during the connection. If both the system and the remote router have PAP authentication, then they negotiate PAP authentication.

Otherwise, the router *always* requests CHAP authentication first; if CHAP is refused, PAP will be negotiated. If the remote end does not accept either PAP or CHAP, the link is dropped; i.e., the router will not communicate without a minimum security level. On the other hand, the router will accept any authentication scheme required by the remote node, including no authentication at all.

During the authentication phase, each side of the link can request authentication using the method they negotiated during LCP.

For CHAP, the router issues a CHAP challenge request to the remote side. The challenge includes the system name and random number. The remote end, using a hash algorithm associated with CHAP, transforms the name and number into a response value. When the remote end returns the challenge response, the router can validate the response challenge value using the entry in the remote router database. If the response is invalid, the call is disconnected. If the other end negotiated CHAP, the remote end can, similarly, request authentication from the local router. The router uses its system name and password to respond to CHAP challenge.

For PAP, when a PAP login request is received from the remote end, the router checks the remote router PAP security using the remote router database. If the remote router is not in the remote router database or the remote router password is invalid, the call is disconnected. If the remote router and password are valid, the local router acknowledges the PAP login request.

If PAP was negotiated by the remote end for the remote-side authentication, the router will issue PAP login requests *only* if it knows the identity of the remote end. The identity is known if the call was initiated from the router, or if the remote end returned a successful CHAP challenge response. For security reasons, the router will *never* identify itself using PAP without first knowing the identity of the remote router.

If PAP was negotiated by the remote end for the local side of the authentication process and the minimum security level is CHAP, as configured in the remote router database, the link will be dropped for a security violation.

# Security Passwords and Levels

When configuring the router, you may set the following passwords:

- System authentication password — the default system password used to access any remote router. Remote sites use this password to authenticate the local site.
- System override password — optional password used only to connect to a specific remote router for authentication by that remote site.
- Remote authentication password — password used by the router to authenticate the remote site. Each remote router entered in the remote router database has a password used when the remote site attempts to gain access to the local router.

To specify a unique system override password for a remote router, use the command **remote SetOurPasswd** (page 203). This password is used instead of the general system password *only* for connecting to a specific remote

router. This allows you to set a unique CHAP or PAP authentication password for authentication of the local site by the remote site *only* when the router connects to that remote site.

A common use is for the system override password is to set a password assigned to you by Internet Service Providers (ISPs). Similarly, the system name of the local router can be overridden for connecting to a specific remote with the command **remote setOurSysName** (page 204).

The router also uses security levels, as follows:

- Remote authentication protocol — Each remote router entered in the remote router database has a minimum security level that must be negotiated before the remote router gains access to the local router.
- System authentication protocol — A system-wide control is available for overriding the minimum security level in the entire remote router database.

# Interoperability Between the Router and Other Equipment

The router uses industry-wide standards to ensure compatibility with routers and equipment from other vendors. To interoperate, the router supports standard protocols on the physical level, data link level for frame type or encapsulation method, and network level. For two systems to communicate directly, they must use the same protocol at each level. Most protocols do not support negotiable options, except for PPP.

The physical protocol level includes hardware and electrical signaling characteristics. This support is provided by the router Ethernet and modem hardware interfaces.

The router supports both ATM (Asychronous Transfer Mode) and Frame Relay transmission. ATM transports fixed-length cells; Frame Relay transports variable-length packets.

The data-link protocol level defines the transmission of data packets between two systems over the LAN or WAN physical link. The frame type or encapsulation method defines a way to run multiple network-level protocols over a single LAN or WAN link. The router supports the following WAN encapsulations:

- PPP (VC multiplexing)
- PPP (LLC multiplexing)
- PPPoE (PPP over Ethernet)
- RFC 1483 (for ATM)
- RFC 1483 with MAC encapsulated routing (for ATM)
- FRF8 (for ATM)
- RFC 1490 (for Frame Relay)
- RFC 1490 with MAC encapsulated routing (for Frame Relay)

# Protocol Conformance

The router conforms to RFCs designed to address performance, authentication, and multi-protocol encapsulation. The following RFCs are supported:

- RFC 1058 Routing Information Protocol (RIP)
- RFC 1144 Compressing TCP/IP headers (Van Jacobson)
- RFC 1220 Bridging Control Protocol (BNCP)
- RFC 1332 IP Control Protocol (IPCP)
- RFC 1334 Password Authentication Protocol and Challenge Handshake Authentication Protocol (PAP/CHAP)
- RFC 1389 RIP2
- RFC 1483 Multiprotocol Encapsulation over ATM Adaptation Layer 5
- RFC 1490 Multiprotocol Interconnect over Frame Relay
- RFC 1542 DHCP Relay Agent
- RFC 1552 Novell IPX Control Protocol (IPXCP)
- RFC 1577 Classical IP and ARP over ATM
- RFC 1631 Network Renumbering
- RFC 1661 Point-to-Point Protocol (PPP)
- RFC 1723 RIP Version 2

- RFC 1877 Automatic IP / DNS
- RFC 1962 PPP Compression Control Protocol (CCP)
- RFC 1969 DES
- RFC 1973 PPP in Frame Relay
- RFC 1974 Stac LZS compression protocol
- RFC 1990 Multi-Link Protocol (MLP)
- RFC 1994 User Authentication PAP / CHAP
- RFC 2104
- RFC 2131 Dynamic Host Configuration Protocol (DHCP)
- RFC 2132 DHCP Client
- RFC 2364 PPP over ATM
- RFC 2419 DES v2
- RFC 2401 Security Architecture for the Internet Protocol
- RFC 2402 IP Authentication Header
- RFC 2403 The Use of HMAC-MD5-96 within ESP and AH
- RFC 2404 The Use of HMAC-SHA-1-96 within ESP and AH
- RFC 2405 The ESP DES-CBC Cipher Algorithm With Explicit IV
- RFC 2406 IP Encapsulating Security Payload (ESP)
- RFC 2407
- RFC 2408
- RFC 2409
- RFC 2410
- RFC 2412
- RFC 2451

## IP Routing

IP routing support, in conformance with RFC 791, provides the ability to process TCP/IP frames at the network layer for routing. IP routing support includes the Routing Interface Protocol (RIP), in conformance with RFC 1058 (RIP v.1) and RFC 1723 (RIP v.2).

## IPX Routing

IPX routing conforms to the Novell® NetWare™ IPX Router Development Guide, Version 1.10.

# Encapsulation Options

This section describes the format of each packet associated with a particular encapsulation option supported by the router.

The encapsulation type for each remote entry is defined using the **remote setProtocol** command (page 204).

## PPP

Each packet begins with a one- or two-byte protocol ID. Typical IDs are:

0xc021 LCP

0x8021 IPCP

0x0021 IP

0x002d Van Jacobson compressed TCP/IP

0x002f Van Jacobson uncompressed TCP/IP

0x8031 Bridge NCP

0x0031 Bridge Frame

The command for this encapsulation option is: **remote setProtocol PPP** <*remoteName*>
See page 204.

**Note:** With PPP over ATM, the address and control fields (i.e., FF03) are never present; this also is the case for LCP packets.

## PPPLLC

This protocol (LLC-multiplexed) allows PPP traffic to be carried simultaneously with other traffic on a single virtual circuit (as opposed to the PPP method of encapsulation—VC multiplexing—which dedicates a virtual circuit to PPP traffic only).

Each PPP packet is prepended with the sequence 0xFEFE03CF. Thus, an LLC packet has the format: 0xFEFE03CF 0xC021.

The command for this encapsulation option is: **remote setProtocol PPPLLC** <*remoteName*>
See page 204.

## RFC 1483 or RFC 1490

### Bridging

User data packets are prepended by the sequence 0xAAAA0300 0x80c20007 0x0000 followed by the Ethernet frame containing the packet.

802.1D Spanning Tree packets are prepended with the header 0xAAAA0300 0x80C2000E.

### Routing

IP packets are prepended with the header 0xAAAA0300 0x00000800.

IPX packets are prepended with the header 0xAAAA0300 0x00008137.

The commands for this encapsulation option are:

**remote setProtocol RFC1483** <*remoteName*> (for ATM)

**remote setProtocol FR** <*remoteName*> (for Frame Relay - RFC 1490)

See page 204.

## MAC Encapsulated Routing: RFC 1483MER (ATM) or RFC 1490MER (Frame Relay)

MER encapsulation allows IP packets to be carried as bridged frames, but does not prevent bridged frames from being sent as well, in their normal encapsulation format: RFC 1483 (ATM) or RFC 1490 (Frame Relay).

If IP routing is enabled, then IP packets are prepended with the sequence 0xAAAA0300 0x80c20007 0x0000 and sent as bridged frames. If IP routing is not enabled, then the packets appear as bridged frames.

The commands for this encapsulation option are:

**remote setProtocol RFC1483MER** <*remoteName*> (for ATM)

**remote setProtocol MER** (for Frame Relay)

## FRF8

IP packets have prepended to them the following sequence: 0x03CC.

The command for this encapsulation option is: **remote setprotocol FRF8** <*remoteName*>

**Note:** This protocol allows sending ATM over Frame Relay.

## rawIP

IP packets do not have any protocol headers prepended to them; they appear as IP packets on the wire. Only IP packets can be transported since there is no possible method to distinguish other types of packets (bridged frames or IPX).

The command for this encapsulation option is: **remote setProtocol rawIP** <*remoteName*>

# System Files

The router's file system is a DOS-compatible file system. Its contents are as follows: :

**SYSTEM.CNF:** These are configuration files containing:

| | |
|---|---|
| DOD | Remote Router Database |
| SYS | System Settings: name, message, authentication method, and passwords |
| ETH | Ethernet LAN configuration settings |

**DHCP.DAT:** DHCP files.

**FILTER.DAT:** Bridge filters.

**KERNEL.F2K:** Router system software (KERNEL.FP1 for IDSL routers).

**ETH.DEF:** File used by the manufacturer to set a default Ethernet configuration.

**ASIC.AIC:** Firmware for the xDSL modem or ATM interface.

**ATM.DAT:** ATM configuration file.

**L2TP Tunneling Database**

**ATOM.DAT**

**SDSL.DAT**

**DMT.DAT**

**IPSEC.DAT**

**IKE.DAT**

**AUTOEXEC.BAT** : Autoexec file of commands to run on next reboot.

**AUTOEXEC.OLD**: Autoexec file that has run already

**Note:** Users should not delete any of these files, unless advised by Tech Support.

Any file contained within the system may be retrieved or replaced using the TFTP protocol. Specifically, configuration files and the operating system upgrades can be updated. Only one copy for the router software is allowed in the router's FLASH memory.

Refer to *Chapter 6. Managing the Router on page 267* for details on software upgrades, booting router software, copying configuration files, and restoring router software to FLASH memory.

# Chapter 2. Planning for Router Configuration

This chapter describes the terminology and the information that you need to obtain before configuring the router. The information needed to configure the router is contingent on the chosen Link Protocol. It is therefore important to know which Link Protocol you are using (this is determined by your Network Service Provider) so that you can refer to the configuration sections that apply to your setup.

When you configure the router using the Command Line Interface, the planning is similar to the process described for Configuration Manager with very few exceptions.

## Important Terminology

You should familiarize yourself with the following terminology as it will be used throughout this chapter.

**Target router.** Router that you are configuring. Also referred to as *local* router**.**

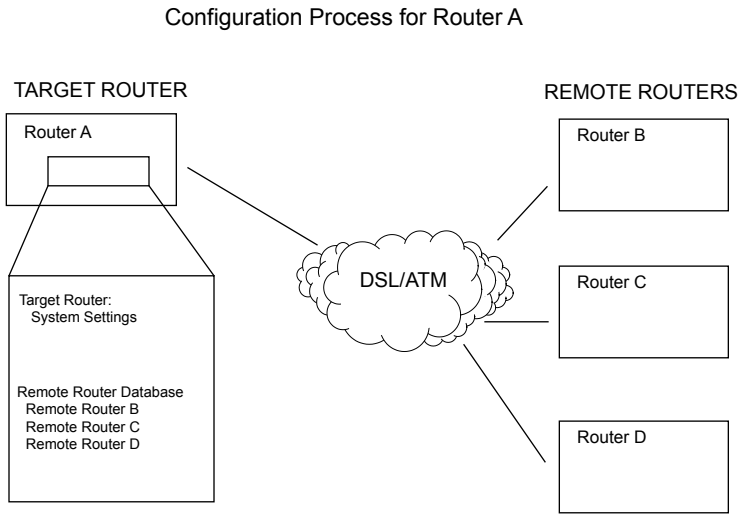
**Remote routers.** All the routers to which the target (local) router may connect.

**Remote router database.** Database which resides in the target router and contains information about the remote routers to which the target router may connect**.**

**Remote router entry.** Entry about a remote router in the target router database. A remote router entry defines:

- Connection parameters
- Security features
- Route addressing and bridging functions

The following diagram illustrates these key words and concepts.



# Essential Configuration Information

This section describes the configuration information associated with each Link Protocol/Network Protocol combination and also provides configuration information for the Dual-Ethernet router.

**If you are using Link and Network Protocols:**

1. Determine which Link Protocol/Network Protocol association you are using from your Network Service Provider (NSP).

2. Select (click) one of the following Link/Network information that applies to your situation:

   **PPP Link Protocol with:**

   - *IP Routing Network Protocol, on page 31*
   - *IPX Routing Network Protocol, on page 33*
   - *Bridging Network Protocol, on page 35*

   **RFC 1483 or RFC 1490 Link Protocol with:**

   - *IP Routing Network Protocol, on page 36*
   - *IPX Routing Network Protocol, on page 37*
   - *Bridging Network Protocol, on page 38*

   **MAC Encapsulated Routing: RFC 1483MER or RFC 1490MER Link Protocol with:**

   - *IP Routing Network Protocol, on page 39*

   **FRF8 Link Protocol with:**

   - *IP Routing Network Protocol, on page 41*

3. Collect the information applicable to your Link/Network Protocol association. This information will be used later in conjunction with the *Configuration Tables* for easy configuration of your router based on your Link/Network protocol. These configuration tables provide step-by-step instructions for a basic configuration for each Link/Network protocol.

   **Note:** Use the blank Network Information Worksheets in Appendix A to collect your network information.

**If you are using a Dual-Ethernet Router:**

Select (click) one of the two following configurations that applies to your situation:

**Configuring the Dual-Ethernet Router as a Bridge, on page 42**

**Configuring the Dual-Ethernet Router for IP Routing, on page 43**

## PPP Link Protocol (over ATM or Frame Relay)

The PPP Link Protocol is an encapsulation method that can be used over ATM (for ATM routers) or Frame Relay (for Frame-Relay routers)

Combined with the IP, IPX, or Bridging Network Protocols, PPP over ATM and PPP over Frame Relay share the same configuration characteristics, except for the connection identifiers: VPI/VCI numbers are used for ATM, and a DLCI number is used for Frame Relay.

Select the Network Protocol that applies to your situation: IP or IPX or Bridging. Collect the information described in the appropriate section. This data will be later used to configure your router using the Command Line Interface commands (see Configuration Tables, on page 45).

### IP Routing Network Protocol

- #### System Names and Authentication Passwords

  ##### For the Target Router

  This information is defined by the user. You must choose a name and authentication password for the target router. They are used by a remote router to authenticate the target router.

  ##### For the Remote Site(s)

  This information is obtained from the Network Service Provider. For each remote site, you *must* have the site name and its authentication password. They are used by the target router to authenticate the remote end. The name and password are used in both PAP and CHAP authentication. Refer to the diagram under PAP/CHAP Security Authentication, on page 20 to see how this information is used.

  **Note 1:** A sample configuration containing names and passwords is provided in the section *Sample Configuration 1: PPP with IP and IPX, on page 59* Chapter 3.

  **Note 2:** If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** <*remoteName*> to disable the authentication.

- #### VPI and VCI Numbers (for ATM routers)

  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- #### DLCI (for Frame Relay routers)

  The DLCI number applies to Frame Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- ## DNS Internet Account Information (optional)

  This information is obtained from your Network Service Provider. Consult with you Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

- ## IP Routing Addresses

  ### For the Ethernet Interface

  This information is defined by the user or your Network Administrator.

  **Ethernet IP Address (local LAN)**
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

  **Note:** An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information. This feature is not normally used, except in special circumstances.

  ### For the WAN Interface

  This information is defined by the Network Service Provider.

  **Source (Target/Local) WAN Port Address**
  If Network Address Translation (NAT) is enabled, you must specify a source WAN IP address for the WAN connection to the remote router if IP address negotiation under PPP does not provide one. Check with your system administrator for details on whether the router must communicate in numbered or unnumbered mode and which addresses are required.

  **Remote WAN Address**
  You may need to specify a remote WAN IP address for the WAN connection to the remote router depending on IP address negotiation under PPP. Check with your system administrator for details on whether the router must communicate in numbered or unnumbered mode and which addresses are required.

  **TCP/IP Remote Routes**
  An IP route includes an IP address, subnet mask, and metric (a number representing the perceived cost in reaching the remote network or station).

  A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. You will need to define the default route to a remote router or, in special circumstances, define an Ethernet gateway. There can be only one default route specified.

- ## System Names and Authentication Passwords

  ### For the Target Router

  This information is defined by the user. You *must* choose a name and authentication password for the target router. They are used by a remote router to authenticate the target router.

  ### For the Remote Site(s)

  This information is obtained from the Network Service Provider. For each remote site, you *must* have the site name and its authentication password. They are used by this target router to authenticate the remote end. The name and password are used in both PAP and CHAP authentication. Refer to the diagrams under *General Security Authentication, on page 17* to see how this information is used.

  **Note 1:** A sample configuration containing names and passwords is provided in the section *Sample Configurations, on page 59*.

  **Note 2:** If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** <*remoteName*> to disable the authentication.

- ## VPI and VCI Numbers

  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- ## DLCI (for Frame-Relay Routers)

  The DLCI number applies to Frame-Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- ## IPX Routing Entries

  These numbers are defined by the Network Administrator. You will need to obtain the following information (most likely from your network administrator) for IPX Routing.

  **Note: IPX routes** define a *path* to a specific destination. They are primarily needed by the routers to allow the servers and clients to exchange packets. A path to a file server will be based on the Internal Network Number of the server. A path to a client will be based on the External Network Number (Ethernet) of the client.

### Internal Network Number

It is a logical network number that identifies an individual Novell server. It is needed to specify a route to the services (i.e., file services, print services) that Novell offers. It must be a unique number.

### External Network (a.k.a. IPX Network Number)

It refers to a physical LAN/wire network segment to which servers, routers, and PCs are connected (Ethernet cable-to-router segment). It must be a unique number.

### WAN Network Number

**Important**: This number is part of the routing information. It is only used to identify the WAN segment between the two routers. Note that only those two routers need to have the WAN Network Number configured.

### Service Advertisement Protocol (SAP)

SAP entries should reflect primary logon servers for the clients on the local LAN. Only the servers on the remote side of the link have to be entered. Local servers do not need to be entered.

### Frame Type

With local servers on your LAN, make sure to select the proper frame type for the IPX network number. To determine this, consult with your network administrator. When you have only NetWare clients on your LAN, keep the default (802.2) selected as most clients can support any type. The frame type choices are:

**802.2** Default recommended by Novell
**802.3** Other most common type
**DIX** For DEC, Intel, Xerox; this setting is also referred to as “Ethernet II”, and it is rapidly becoming obsolete.

**Note:** For step-by-step information on how to configure IPX routing, see Configuring IPX Routing, on page 309.

## Bridging Network Protocol

- ### System Names and Authentication Passwords

  #### For the Target Router

  This information is defined by the user. You *must* choose a name and authentication password for the target router. They are used by a remote router to authenticate the target router.

- ### For the Remote Site(s)

  This information is obtained from the Network Service Provider. For each remote site, you *must* have the site name and its authentication password. They are used by the target router to authenticate the remote end. The name and password are used in both PAP and CHAP authentication. Please refer to the diagram under *General Security Authentication, on page 17* to see how this information is used.

  **Note 1:** A sample configuration containing Names and Passwords is provided in the section Sample Configuration 1: PPP with IP and IPX, on page 59.

  **Note 2:** If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** *<remoteName>* to disable the authentication.

- ### VPI and VCI Numbers

  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- ### DLCI (for Frame-Relay Routers)

  The DLCI number applies to Frame-Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- ### DNS Internet Account Information (optional)

  This information is obtained from the Network Service Provider. Consult with you Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

  **Note:** If you intend to connect to the Internet only, enter this information using the Internet Quick Start configurator.

# RFC 1483/RFC 1490 Link Protocols

The Link Protocol RFC 1483 is a multiprotocol encapsulation method over ATM and is used by ATM routers. RFC 1490 is a multiprotocol encapsulation method over Frame-Relay and is used by Frame-Relay routers.

RFC 1483 and RFC 1490 combined with the IP, IPX, or Bridging Network Protocols share the same configuration characteristics, except for the connection identifiers: VPI/VCI numbers are used for RFC 1483 and a DLCI number is used for RFC 1490.

Obtain the information as described in the appropriate section. This data will be used later to configure your router using the Command Line Interface (see Configuration Tables, on page 45).

## IP Routing Network Protocol

- ### VPI and VCI Numbers (for RFC 1483)

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- ### DLCI (for RFC 1490)

  The DLCI number applies to Frame-Relay routers only.  Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- ### DNS Internet Account Information (optional)

  This information is obtained from the Network Service Provider.  Consult with you Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

- ### IP Routing Entries

  #### For the Ethernet Interface

  This information is defined by the user or the Network Administrator.

  **Ethernet IP Address (Local LAN)**
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

**TCP/IP Ethernet Routes**
You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information.

For the WAN Interface

This information is obtained from the Network Administrator.

**Source (Target/Local) WAN Port Address**
If Network Address Translation (NAT) is enabled, you must specify a source WAN IP address for the WAN connection to the remote router. Check with your system administrator for details.

If NAT is not enabled, you may need to specify a source WAN IP address for the WAN connection to the remote router. Check with your system administrator for details.

**TCP/IP Remote Route**s
An IP route includes an IP address, subnet mask, and metric (a number representing the perceived cost in reaching the remote network or station).

A **TCP/IP Default Route** default route should be designated in the routing table for all traffic that cannot be directed to other specific routes. You will need to define the default route to a remote router or, in special circumstances, define an Ethernet gateway. There can be only one default route specified.

## IPX Routing Network Protocol

- ### VPI and VCI Numbers (for RFC 1483)

  The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- ### DLCI (for RFC 1490)

  The DLCI number applies to Frame Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- ### IPX Routing Entries

  The user or the Network Administrator defines this information.

  **Note: IPX routes** define a *path* to a specific destination. They are primarily needed by the routers to allow the servers and clients to exchange packets. A path to a file server will be based on the Internal Network Number of the server. A path to a client will be based on the External Network Number (Ethernet) of the client.

#### Internal Network Number

This is a logical network number that identifies an individual Novell server. It is needed to specify a route to the services (i.e., file services, print services) that Novell offers. It must be a unique number.

#### External Network (a.k.a. IPX Network Number)

This number refers to a physical LAN/wire network segment to which servers, routers, and PCs are connected (Ethernet cable-to-router segment). It must be a unique number.

#### WAN Network Number

**Important:** This number is part of the routing information. It is only used to identify the WAN segment between the two routers.

**Note:** Only the two routers need to have the WAN Network Number configured.

#### Service Advertisement Protocol (SAP)

SAP entries should reflect primary logon servers for the clients on the local LAN. Only the servers on the remote side of the link have to be entered. Local servers do not need to be entered.

#### Frame Type

With local servers on your LAN, make sure to select the proper frame type for the IPX network number. To determine this, consult with your network administrator. When you have only NetWare clients on your LAN, keep the default (802.2) selected as most clients can support any type.
The frame type choices are:

**802.2** Default recommended by Novell
**802.3** Other most common type
**DIX** For DEC, Intel, Xerox; this setting is also referred to as “Ethernet II”, and it is rapidly becoming obsolete.

## Bridging Network Protocol

- ### VPI and VCI Numbers (with RFC 1483)

The VPI and VCI numbers apply to ATM routers only. Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- ### DLCI (with RFC 1490)

The DLCI number applies to Frame-Relay routers only. Your Network Service Provider or your Network Access Provider will provide you with a Data Link Connection Identifier (DLCI). The DLCI is an address identifying your connection.

- DNS Internet Account Information (optional)

  This information is obtained from the Network Service Provider. Consult with your Network Service Provider to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

## MAC Encapsulated Routing

MAC Encapsulated Routing (MER) allows IP packets to be carried as bridged frames (bridged format). The Link Protocol RFC 1483 with MER (referred to as RFC 1483MER) is a multiprotocol encapsulation method over ATM used by ATM routers. RFC 1490 with MER (referred to as RFC 1490MER) is a multiprotocol encapsulation method over Frame Relay used by Frame-Relay routers.

RFC 1483MER and RFC 1490MER combined with the IP, IPX, or Bridging Network Protocols share the same configuration characteristics, except for the connection identifiers: VPI/VCI numbers are used for RFC 1483MER and a DLCI number is used for RFC 1490.

Obtain the information as described in the appropriate section. This data will be later used to configure your router using the Command Line Interface (see Configuration Tables, on page 45).

### IP Routing Network Protocol

- VPI and VCI Numbers (for RFC 1483MER)

  The VPI and VCI numbers apply to ATM routers only.  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- DLCI (for RFC 1490MER)

  The DLCI number applies to Frame Relay routers only.  Your Network Service Provider or your Network Access Provider will provide you with a DLCI (Data Link Connection Identifier). The DLCI is an address identifying your connection.

- DNS Internet Account Information (optional)

  This information is obtained from the Network Service Provider. Consult with your Network Service Provider to find out if you need to enter the following information:

  - DNS server address

- DNS second server address
- DNS domain name

**Note:** If you intend to only connect to the Internet, enter this information using the Internet Quick Start configurator.

## • IP Routing Entries

### For the Ethernet Interface

This information is defined by the user or the Network Administrator.

**Ethernet IP Address (Local LAN)**
An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

**TCP/IP Ethernet Routes**
You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information between them.

### For the ATM WAN Interface

This information is obtained from the Network Administrator or the Network Service Provider.

**Source (Target/Local) WAN Port Address and Mask**
You must specify a Source WAN IP address for the WAN connection to the remote router (whether or not Network Address Translation is enabled). The Source WAN Address is the address of the local router on the remote network. The mask is the mask used on the remote network. Check with your system administrator for details.

**TCP/IP Remote Routes**
If you are using RFC 1483MER or RFC 1490MER, the IP route includes an IP address, subnet mask, metric (a number representing the perceived cost in reaching the remote network or station), and a *gateway*. The gateway address that you enter is the address of a router on the remote LAN. Check with your system administrator for details.

A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes. You will need to define the default route to a remote router or, in DLCI ( special circumstances, define an Ethernet gateway. There can be only one default route specified.

## FRF8 Link Protocol

The FRF8 Link Protocol is an encapsulation method that allows an ATM router to interoperate with a Frame- Relay network.

FRF8 is only used in conjunction with the IP Network Protocol. Obtain the information described below. This data will be used later to configure your router using the Command Line Interface (see Configuration Tables, on page 45).

### IP Routing Network Protocol

- #### VPI and VCI Numbers

  Your router may have been preconfigured with VPI/VCI numbers. If not, you will have to obtain these numbers from your Network Service Provider and then configure them.

  If you are connecting to multiple remote sites, you will need to obtain additional VPI and VCI numbers from your Network Service Provider and/or Network Access Provider. These numbers identify the remote destination and must, therefore, be unique for each remote.

- #### DNS Internet Account Information (optional)

  The following information is obtained from the Network Service Provider. Consult with your NSP to find out if you need to enter the following information:

  - DNS server address
  - DNS second server address
  - DNS domain name

  **Note:** If you intend to connect only to the Internet, enter this information using the Internet Quick Start configurator.

- #### IP Routing Entries

  ##### For the Ethernet Interface

  This information is defined by the user or the Network Administrator.

  **Ethernet IP Address (Local LAN)**
  An Ethernet LAN IP address and subnet mask are required for the router's local Ethernet LAN connection.

  **TCP/IP Ethernet Routes**
  You normally do not need to define an Ethernet IP route. An Ethernet IP route consists of an IP address, a mask, a metric, and a gateway. An Ethernet route is usually defined when there are multiple routers on the Ethernet that cannot exchange routing information.

For the ATM WAN Interface

This information is obtained from the Network Administrator or the Network Service Provider.

**Source (Target/Local) WAN Port Address and Mask**
You must specify a Source WAN IP address for the WAN connection to the remote router (whether or not Network Address Translation is enabled. The Source WAN address is the address of the local router on the remote network. The mask is the mask used on the remote network. Check with your system administrator for details.

**TCP/IP Remote Routes**
If you are using FRF8, the IP route includes an IP address, subnet mask, metric (a number representing the perceived cost in reaching the remote network or station). Check with your system administrator for details.

A **TCP/IP Default Route** should be designated in the routing table for all traffic that cannot be directed to other specific routes.
You will need to define the default route to a remote router or, in special circumstances, define an Ethernet gateway. There can be only one default route specified.

# Dual-Ethernet Router Configuration

## General Information on Dual Ethernet router

To configure the Dual-Ethernet router, access the router using the Command Line Interface (CLI). The CLI can be accessed from a Telnet or a console session (using the console cable) connected to the router's default IP address of 192.169.254.254. You can also configure the router using the Web browser GUI. Refer to the *Dual-Ethernet Router Quick Start Guide*.

The Dual-Ethernet router has two interfaces:

ETH/0—refers to the router's hub with four 10Base-T connectors

ETH/1—refers to the single 10Base-T connector (or the second hub on the new hardware model)

Bridging is enabled by default when the router boots up. IP and IPX routing are disabled.

The router's default IP address is 192.168.254.254.

DHCP is enabled by default and the router's DHCP server issues IP addresses to any PC request. The DHCP default IP pool is 192.168.254. 2 through 192.168.254.20.

To connect to the router, use the router's default IP address using a Telnet session, for example, and any 10Base-T port on the router.

**Warning:** You cannot boot from the ETH/1 interface.

## Configuring the Dual-Ethernet Router as a Bridge

This router is configured by default as a bridge and no configuration steps are needed. The user needs only establish a connection to the remote location (to the Internet Service Provider, for example).

Bridging is enabled by default when the router boots up. IP and IPX routing are disabled.

## Configuring the Dual-Ethernet Router for IP Routing

The **eth** commands are used to configure the Dual-Ethernet router for IP routing. Refer to the section *Dual-Ethernet Router Commands (ETH), on page 213*, for usage and syntax information.

The last argument of each ETH command determines which interface is being configured (0 for ETH/0, 1 for ETH/1).

Each interface (ETH/0 and ETH/1) must be set. A minimum of one route must be defined to have a working configuration. This is generally a default route on the ETH/1 interface where all traffic otherwise specified is automatically forwarded. This default route is: 0.0.0.0 255.255.255.255 1.

The Gateway address is the IP address supplied by your Internet Service Provider or Network Administrator.

You can customize your router by using the scripting feature, which loads batch files of preset configuration commands into the router (refer to the *Batch File Command Execution, on page 279* section).

A Dual-Ethernet router sample configuration with IP Routing is provided in the *Sample Configuration 3: Configuring a Dual-Ethernet Router for IP Routing, on page 73* section.

# Chapter 3. Configuring Router Software

This chapter covers configuration tables and verifying the router configuration. It also provides sample configurations.

Configuration commands are outlined for each **Link Protocol/Network Protocol** supported by the router. The information needed to configure the router is contingent on the chosen Link Protocol. It is therefore important to know which Link Protocol you are using (this is determined by your Network Service Provider) to be able to refer to the configuration sections that apply to your setup.

A configuration table for the Dual-Ethernet Router (with IP routing enabled) is also provided.

The section on verifying the router configuration describes how to test IP, IPX, or Bridging.

In this chapter, you will find two sample configurations with diagrams, commands, and list outputs.

**Note 1:** For usage conventions and a complete description of the commands mentioned in this chapter, refer to Chapter 5. Command Line Interface Reference on page 135.

**Note 2:** To configure the router software, the Command Line Interface is available to you at all times after you have installed the router hardware, connected to the router with a terminal-emulation session (or ASCII terminal), and powered the unit on. This chapter assumes that you have successfully installed the router hardware as described in the *Quick Start Guide*.

If you intend to use the Command Line Interface through Configuration Manager, it is assumed that you have installed the Configuration Manager software and can access the terminal window (refer to the *Quick Start Guide*).

**Note 3:** Worksheets are provided in Appendix A so that you can enter details about your target router and remote routers. The worksheets list the commands associated with setting the features.

To configure the target router, you need to fill out one chart for the target router and one remote router chart for *each* remote router to be entered into the remote router database.

If you are setting up both ends of the network, you will need a *mirror image* of the information listed below for configuring the router on the other end of the link.

**Important:** If you change any the of the following settings, you must use the commands **reboot** and **save** for the changes to take effect:

**Ethernet LAN:** Ethernet IP or IPX address, TCP/IP routing, IPX routing

**Bridging:** Bridging, Filters

**Remote Router:** TCP/IP route addresses, IPX routes, IPX SAPs and bridging control, and enable, disable, or add remote routers

# Configuration Tables

The following tables give you step-by-step instructions for standard configurations of the following Network Protocol/Link Protocol associations, as well as a configuration table for a Dual-Ethernet Router:

- PPP Link Protocol with IP Routing Network Protocol
- PPP Link Protocol with IPX Routing Network Protocol
- PPP Link Protocol with Bridging Network Protocol
- RFC 1483/RFC 1490 Link Protocols with IP Routing Network Protocol
- RFC 1483/RFC 1490 Link Protocols with IPX Routing Network Protocol
- RFC 1483/RFC 1490 Link Protocols with Bridging Network Protocol
- RFC 1483MER/RFC 1490MER Link Protocols with IP Network Protocol
- FRF8 Link Protocol with IP Routing Network Protocol
- Mixed Network Protocols (combinations of two or three network protocols)
- Dual-Ethernet Router with IP routing

**Note:** Blank Network Configuration Worksheets are available in Appendix A.

Using the tables:

1. Find the configuration table that fits your particular Network Protocol/Link Protocol association. These tables are designed to provide easy step-by-step instructions.
2. Use the blank Network Configuration Worksheets provided in Appendix A to enter the commands in the order that they are given in the **Commands** column of the configuration tables.
3. You may want to refer to the sample configurations at the end of this chapter.

## Configuring PPP with IP Routing

This table outlines configuration commands for the PPP Link Protocol with the IP Routing Network Protocol.

| **PPP with IP Routing** | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Name | Required | **system name <***name***>** |
| System Message | Optional | **system msg <***message***>** |
| Authentication Password | Required | **system passwd <***password***>** |
| Ethernet IP Address | As required | **eth ip addr <***ipaddr***> <***ipnetmask***>** [<*port#*>] |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC[a]<br>(for ATM routers) | Select: PPP<br>Enter: VPI/VCI numbers | **remote setProtocol PPP <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName*>** |
| Link Protocol/DLCI[b]<br>(for Frame Relay routers) | Select: PPP<br>Enter: DLCI number | **remote setProtocol PPP <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Security [c]<br>Remote's Password | Choose security level<br>Enter: password | **remote setAuthen <***protocol*> **<***remoteName***>**<br>**remote setOurPasswd <***password***> <***remoteName***>** |
| Bridging On/Off | Must be off | **remote disBridge <***remoteName***>** |
| TCP/IP Route Address | Enter: Explicit or default route | **remote addIproute <***ipnet***> <***ipnetmask***> <***hops***>**<br>**<***remoteName***>** |
| If NAT is enabled: | To enable NAT, use: | **remote setIpTranslate** on **<***remoteName***>** |
| | You may need to enter a Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***> <***mask***>**<br>**<***remoteName***>** |
| If NAT is not enabled: | You may need to enter a Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***>** <*mask***>**<br>**<***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing<br>(Internet Firewall) | Must be enabled<br>(optional) | **eth ip enable**<br>**eth firewall <**on | off**>** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using PPP in an ATM environment.

b Enter this information if you are using PPP in a Frame Relay environment.

c If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen** <*remoteName*> to disable the authentication.

## Configuring PPP with IPX Routing

This table outlines configuration commands for the PPP Link Protocol with the IPX Routing Network Protocol.

**Note:** Appendix B provides step-by-step information on how to configure IPX routing.

| PPP with IPX Routing | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Name | Required | **system name <***name***>** |
| System Message | Optional | **system msg <***message***>** |
| Authentication Password | Required | **system passwd <***password***>** |
| Ethernet IP Address | As required | **eth ip addr <***ipaddr***> <***ipnetmask***>**[<*port#*>] |
| Settings DHCP | Already enabled; addit. settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <** *ipaddr* **>** |
| Change Login | Optional | **system admin <***password***>** |
| Ethernet IPX Network # | Enter: IPX network #<br>Frame Type (default: 802.2) | **eth ipx addr <***ipxnet***>** [<*port#*>]<br>**eth ipx frame <***type***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC[a]<br>(for ATM routers) | Select: PPP<br>Enter: VPI/VCI numbers | **remote setProtocol PPP <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** |
| Link Protocol/DLCI[b]<br>(for Frame Relay routers) | Select: PPP<br>Enter: DLCI number | **remote setProtocol** PPP **<***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Security[c]<br>Remote's Password | Choose security level<br>Enter: password | **remote setAuthen <***protocol***> <***remoteName***>**<br>**remote setPasswd <***password***> <***remoteName***>** |
| Bridging On/Off | Must be off | **remote disBridge <***remoteName***>** |
| IPX Routes<br>Add | Enter appropriate info | **remote addIpxroute** <*ipxNet*> **<***metric***> <***ticks***>**<br>**<***remoteName*> |
| IPX SAPs<br>Add | Enter appropriate info | **remote addIpxsap <***servicename***> <***ipxNet***>**<br>**<***ipxNode***> <***socket***> <***type***> <***hops***> <***remoteName***>** |
| WAN Network # | Enter appropriate info | **remote setIpxaddr <***ipxNet***> <***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing | Must be disabled | **eth ip disable** |
| IPX Routing | Must be enabled | **eth ipx enable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using PPP in an ATM environment.
b Enter this information if you are using PPP in a Fram- Relay environment.
c If the ISP does not support the authentication of the ISP system by the caller, use the command:
**remote disauthen** <*remoteName*> to disable the authentication.

## Configuring PPP with Bridging

This table outlines configuration commands for the PPP Link Protocol with the Bridging Network Protocol.

| PPP with Bridging | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Name | Required | **system name <***name***>** |
| System Message | Optional | **system msg <***message***>** |
| Authorization Password | Required | **system passwd <***password***>** |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <** *ipaddr* **>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC[a] (for ATM routers) | Select: PPP<br>Enter: VPI/VCI | **remote setProtocol PPP <***remoteName***>**<br>**remote setPVC** *<vpi number>*<vci number>* **<***remoteName***>** |
| Link Protocol/DLCI[b] (for Frame Relay routers) | Select: PPP<br>Enter: DLCI number | **remote setProtocol PPP <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Security[c]<br>Remote's Password | Choose security level<br>Enter: Password | **remote setAuthen <***protocol***> <***remoteName***>**<br>**remote setOurPasswd <***password***> <***remoteName***>** |
| Bridging On/Off | Must be ON | **remote enaBridge <***remoteName***>** |
| **IP and IPX Routing** | | |
| IP Routing | Must be disabled | **eth ip disable** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using PPP in an ATM environment.

b Enter this information if you are using PPP in a Frame-Relay environment.

c If the ISP does not support the authentication of the ISP system by the caller, use the command **remote disauthen <***remoteName***>** to disable the authentication.

## Configuring PPP over Ethernet (PPPoE)

This table outlines configuration commands for the PPP Link Protocol with the Bridging Network Protocol over Ethernet.

| PPP over Ethernet | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Name | Required | **system name <***name***>** |
| System Message | Optional | **system msg <***message***>** |
| Authentication Password | Required | **system passwd <***password***>** |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <** *ipaddr* **>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Set PPPoE Service | Enter service name or * for any service | **remote setpppoeservice** <service> <remoteName> |
| Link Protocol/PVC[a] (for ATM routers) | Select: PPP<br>Enter: VPI/VCI | **remote setProtocol PPP <***remoteName***>**<br>**remote setPVC** *<vpi number>****<***vci number***> <***remoteName***>** |
| Link Protocol/DLCI[b] (for Frame Relay routers) | Select: PPP<br>Enter: DLCI number | **remote setProtocol PPP <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Security | Enter system name and password; disable authentication | **remote setOurSysName <***name***> <***remoteName***>**<br>**remote setOurPasswd <***password***> <***remoteName***>**<br>**remote disauthen <***remoteName***>** |
| Bridging On/Off | Must be ON | **remote enaBridge <***remoteName***>** |
| **IP and IPX Routing** | | |
| IP Routing | If desired, enable and then define routes and filters. | **eth ip enable**<br>**remote addiproute . . . <***remoteName***>**<br>**remote bindIPVirtualRoute . . . <***remoteName***>**<br>**remote setiptranslate on <***remoteName***>** |
| IPX Routing | If desired, set address | **remote setipxaddr . . . <***remoteName***>** |
| **Bridging through PPPoE Remote** | | |
| Remote Bridge | | **remote enabridge <***remoteName***>**<br>**remote addbridge * <***remoteName***>** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using PPP in an ATM environment.
b Enter this information if you are using PPP in a Frame-Relay environment.

## Configuring RFC 1483 / RFC 1490 with IP Routing

This table outlines configuration commands for the RFC 1483 and the RFC 1490 Link Protocols with the IP Routing Network Protocol.

| RFC 1483 / RFC 1490 with IP Routing | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Message | Optional | **system msg <***message***>** |
| Ethernet IP Address | As required | **eth ip addr <***ipaddr***> <***ipnetmask***>** [**<***port#***>**] |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC[a] (for ATM routers) | Select: RFC 1483<br>Enter: VPI/VCI Numbers | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** |
| Link Protocol/DLCI[b] (for Frame Relay routers) | Select: FR<br>Enter: DLCI number | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Bridging On/Off | Must be OFF | **remote disBridge <***remoteName***>** |
| TCP/IP Route Address | Enter: Explicit or default route with remote gateway | **remote addiproute <***ipnet***> <***ipnetmask***> <***hops***>**<br>**<***remoteName***>** |
| If Address Translation (NAT) is enabled: | To enable NAT, use: | **remote setIpTranslate on <***remoteName***>** |
| TCP/IP Route Addresses | Enter: Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***> <***mask***> <***remoteName***>** |
| If NAT is off: TCP/IP Route Addresses | You may still need to enter a Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***> <***mask***> <***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing (Internet Firewall) | Must be enabled (Optional) | **eth ip enable**<br>**eth firewall <on \| off >** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using RFC 1483 in an ATM environment.

b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

## Configuring RFC 1483 / RFC 1490 with IPX Routing

This table outlines configuration commands for the RFC 1483 and RFC 1490 Link Protocols with the IPX Routing Network Protocol.

**Note:** Appendix B provides step-by-step information on how to configure IPX routing.

| **RFC 1483 / RFC 1490 with IPX Routing** | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Message | Optional | **eth ip addr <***ipaddr***> <***ipnetmask***>** [<*port#*>] |
| Ethernet IP Address | As required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <** *ipaddr* **>** |
| DHCP Settings | Already enabled; additional settings may be required | **eth ipx addr <***ipxnet***>** [<*port#*>] |
| Ethernet IPX Network # | Enter: IPX Network # Frame Type (default is 802.2) | **eth ipx frame <***type***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC<br>(for ATM routers) | Select: RFC 1483<br>Enter: VPI/VCI Numbers | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <***vpi number***>* <***vci number***>**<br>**<***remoteName***>** |
| Link Protocol/DLCI[a]<br>(for Frame Relay routers) | Select: FR<br>Enter: DLCI number | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <** *number***> <***remoteName***>** |
| Bridging on/off | Must be off | **remote disBridge <***remoteName***>** |
| IPX Routes<br>Add | Enter appropriate info | **remote addIpxroute <***ipxNet***> <***metric***> <***ticks***>**<br>**<***remoteName***>** |
| IPX SAPs<br>Add | Enter appropriate info | **remote addIpxsap <***servicename***> <***ipxNet***>**<br>**<** *ipxNode***> <***socket***> <***type***> <***hops***> <***remoteName*>** |
| WAN Network Number | Enter appropriate info | **remote setIpxaddr <***ipxNet***> <***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing<br>(Internet Firewall) | Must be disabled<br>(optional) | **eth ip disable**<br>**eth firewall** <on \| off > |
| IPX Routing | Must be enabled | **eth ipx enable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using RFC 1490 in a Frame Relay environment.

## Configuring RFC 1483 / RFC 1490 with Bridging

This table outlines configuration commands for the RFC 1483 and RFC 1490 Link Protocols with the Bridging Network Protocol.

| RFC 1483 / RFC 1490 with Bridging | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Message | Optional | **system msg <***message***>** |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC (for ATM routers) | Select: RFC 1483<br>Enter: VPI/VCI Numbers | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** |
| Link Protocol/DLCI[a] (for Frame Relay routers) | Select: FR<br>Enter: DLCI number | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Bridging On/Off | Must be on | **remote enaBridge <***remoteName***>** |
| **IP and IPX Routing** | | |
| IP Routing | Must be disabled | **eth ip disable** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using RFC 1490 in a Frame-Relay environment.

## Configuring MAC Encapsulated Routing: RFC 1483MER / RFC 1490MER with IP Routing

This table outlines configuration commands for the RFC 1483MER and RFC 1490MER Link Protocols with the IP Routing Network Protocol.

| **RFC 1483MER / RFC 1490 MER with IP Routing** | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Message | Optional | **system msg <***message***>** |
| Ethernet IP Address | As required | **eth ip addr <***ipnet***> <***ipnetmask***>** [**<***port#***>**] |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC[a] (for ATM routers) | Select: RFC 1483MER<br>Enter: VPI/VCI Numbers | **remote setProtocol** RFC1483MER **<***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** |
| Link Protocol/DLCI[b] (for Frame Relay routers) | Select: MER<br>Enter: DLCI number | **remote setProtocol MER <***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** |
| Bridging On/Off | Must be off | **remote disBridge <***remoteName***>** |
| TCP/IP Route Address | Enter: Explicit or default route with remote gateway | **remote addiproute <***ipnet***> <***ipnetmask***><***ipGateway***>**<br>**<***ipGateway***> <***remoteName***>** |
| If NAT is enabled: | To enable NAT, use: | **remote setIpTranslate on <***remoteName***>** |
| If NAT is OFF: | Enter: Source WAN Port Address + mask of the remote network | **remote setSrcIpAddr <***ipaddr***> <***mask***><***remoteName***>** |
| TCP/IP Route Addresses | Enter a Source WAN Port Address + mask of the remote network's mask | **remote setSrcIpAddr <***ipaddr***> <***mask***> <***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing (Internet Firewall) | Must be enabled (optional) | **eth ip enable**<br>**eth firewall <**on | off **>** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

a Enter this information if you are using RFC 1483 in an ATM environment.

b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

## Configuring FRF8 with IP Routing

This table outlines configuration commands for the FRF8 Link Protocol with the IP Routing Network Protocol.

| **FRF8 with IP Routing** | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Message | Optional | **system msg <***message***>** |
| Ethernet IP Address | As required | **eth ip addr <***ipaddr***> <***ipnetmask***>** [<*port#*>] |
| DHCP Settings | Already enabled; additional settings may be required | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** |
| Change Login | Optional | **system admin <***password***>** |
| **Remote Routers** | | |
| New Entry | Enter: Remote Name | **remote add <***remoteName***>** |
| Link Protocol/PVC | Select: FRF8<br>Enter: VPI/VCI Numbers | **remote setProtocol FRF8 <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** |
| Bridging On/Off | Must be off | **remote disBridge <***remoteName***>** |
| TCP/IP Route Address | Enter: explicit or default route | **remote addIproute <***ipnet***> <***ipnetmask*> **<***hops***>**<br>**<***remoteName***>** |
| If Address Translation (NAT) is enabled: | To enable NAT, use: | **remote setIpTranslate on <***remoteName***>** |
| If NAT is OFF: | Enter: Source WAN Port Address + mask of the remote network | **remote setSrcIpAddr <***ipaddr***> <***mask***><***remoteName***>** |
| TCP/IP Route Addresses | Enter a Source WAN Port Address + mask of the remote network | **remote setSrcIpAddr <***ipaddr***> <***mask***><***remoteName***>** |
| **IP and IPX Routing** | | |
| TCP/IP Routing (Internet Firewall) | Must be enabled (Optional) | **eth ip enable**<br>**eth firewall <**on | off **>** |
| IPX Routing | Must be disabled | **eth ipx disable** |
| **Store**<br>**Reboot** | | **save**<br>**reboot** |

## Configuring Mixed Network Protocols

Several network protocols can be configured concurrently in the same router. The possible combinations are:

- Bridging + IP routing
- Bridging + IPX routing
- Bridging + IP routing + IPX routing
- IP routing + IPX routing

**General configuration rules**:

- IP (and IPX) routing takes precedence over bridging.
- Each network protocol in the combination is individually configured as described in the preceding tables.
- When configuring multiple network protocols, *make sure that they are all enabled* (even though the preceding individual configuration tables show them to be mutually exclusive).

**Example**:

To configure bridging + IP routing (both with Link Protocol RFC 1483), refer to the preceding *RFC 1483 with Bridging* and *RFC 1483 with IP Routing* tables. Follow the instructions described in the tables, except for the Bridging and IP Routing settings. Since you are configuring both bridging and IP routing, make sure that these two protocols are both enabled (even though the individual configuration tables you are referring to are showing them to be mutually exclusive). Configure Bridging and then IP Routing. Remember that IP Routing has precedence over Bridging.

## Configuring a Dual-Ethernet Router for IP Routing

This table outlines commands used to configure a Dual-Ethernet router for IP routing.

| **Dual-Ethernet Router with IP Routing** | | |
|---|---|---|
| **Steps** | **Settings** | **Commands** |
| **System Settings** | | |
| System Name<br>**System Settings** | Optional | **system name** <*name*> |
| Message | Optional | **system msg <***message***>** |
| **Ethernet Settings** | | |
| Routing/ Bridging Controls | Enable IP routing<br>Disable bridging | **eth ip enable**<br>**eth br disable** |
| ETH/0 IP Address | Define ETH/0 IP address for the hub side | **eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>] |
| ETH/1 IP Address | Define ETH/1 IP address for the single 10Base-T side | **eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>] |
| TCP/IP default route address | ETH/0 sends all traffic to ETH/1 | **eth ip addroute** <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*port#*>] |
| **DHCP Settings**<br>Already enabled; additional settings may be required | | |
| DHCP Settings | Define DHCP network for ETH/1 | **dhcp add | <***net*> <*mask*> | <*ipaddr*> | <*code*> <*min*> <*max*> <*type*> |
| | Create an address pool for ETH/1 | **dhcp set addresses** <*first ipaddr*> <*last ipaddr*> |
| | DNS Domain Name | **dhcp set valueoption domainname <***domainname***>** |
| | DNS Server | **dhcp set valueoption domainnameserver** <*ipaddr*> |
| | WINS Server Address | **dhcp set valueoption winsserver** <*ipaddr*> |

# Verify the Router Configuration

## Test IP Routing

### Test IP Routing over the Local Ethernet LAN (from PC)

- Use the TCP/IP **ping** command or a similar method to contact the configured target router specifying the Ethernet LAN IP address.
- If you cannot contact the router, verify that the Ethernet IP address and subnet mask are correct and check the cable connections.
- Make sure that you have saved and rebooted after setting the IP address.
- Check Network TCP/IP properties under Windows 95. If you are running Windows 3.1, check that you have a TCP/IP driver installed.

### Test IP Routing to a Remote Destination

- Using the TCP/IP **ping** command, contact a remote router from a local LAN-connected PC. When you enter the **ping** command, the router will connect to the remote router using the DSL line.
- If remote or local WAN IP Addresses are required, verify that they are valid.
- Use the **iproutes** command to check, first, the contents of the IP routing table and, second, that you have specified a default route as well.

### Test Routing from a Remote Destination

- Have a remote router contact the target router using a similar method.

### Test TCP/IP Routes

- Contact a station, subnetwork, or host located on the network beyond a remote router to verify the TCP/IP route addresses entered in the remote router database.
- Verify that you configured the correct static IP routes.
- Use the **iproutes** command to check the contents of the IP routing table.

## Test Bridging to a Remote Destination

Use any application from a local LAN-attached station that accesses a server or disk using a protocol that is being bridged on the remote network beyond the remote router. If you cannot access the server:

- Verify that you have specified a default destination remote router.
- Make sure that you have enabled bridging to the remote router.
- Check that bridge filtering does not restrict access from the local station.

## Test IPX Routing

One way to test IPX routing is to check for access to servers on the remote LAN. Under Windows, use the **NetWare Connections** selection provided with NetWare User Tools. Under DOS, use the command **pconsole** or type **login** on the login drive (usually F:). Select the printer server and verify that the server you have defined is listed. When you attempt to access the server, the router will connect to the remote router using the DSL line.

If you cannot access the remote server:

- Check that the local Ethernet LAN IPX network number is correct.
- Verify that the WAN link network number is the same as the remote WAN link network number.
- Check cable connections and pinouts.
- Verify that the IPX routes and IPX SAPs you have specified are correct.
- List the contents of the routing and services tables using the **ipxroutes** and **ipxsaps** commands, respectively.
- Make sure that the security authentication method and password that you configured match the remote router.

# Sample Configurations

## Sample Configuration 1: PPP with IP and IPX

This configuration example comprises:

- A scenario describing the configuration
- A diagram showing the configuration of the SOHO router
- Tables containing the configuration settings for this example
- Several **list** command outputs that are used to check the information entered for this particular configuration
- Information about the names and passwords that are used in this configuration example (required for PPP)

**Note:** Appendix A contains blank Network Information Worksheets; use them to fill in the information for your own configuration. If you selected “install documentation”, these samples and others are copied from the installation CD into the samples directory where the GUI was installed .

### Scenario:

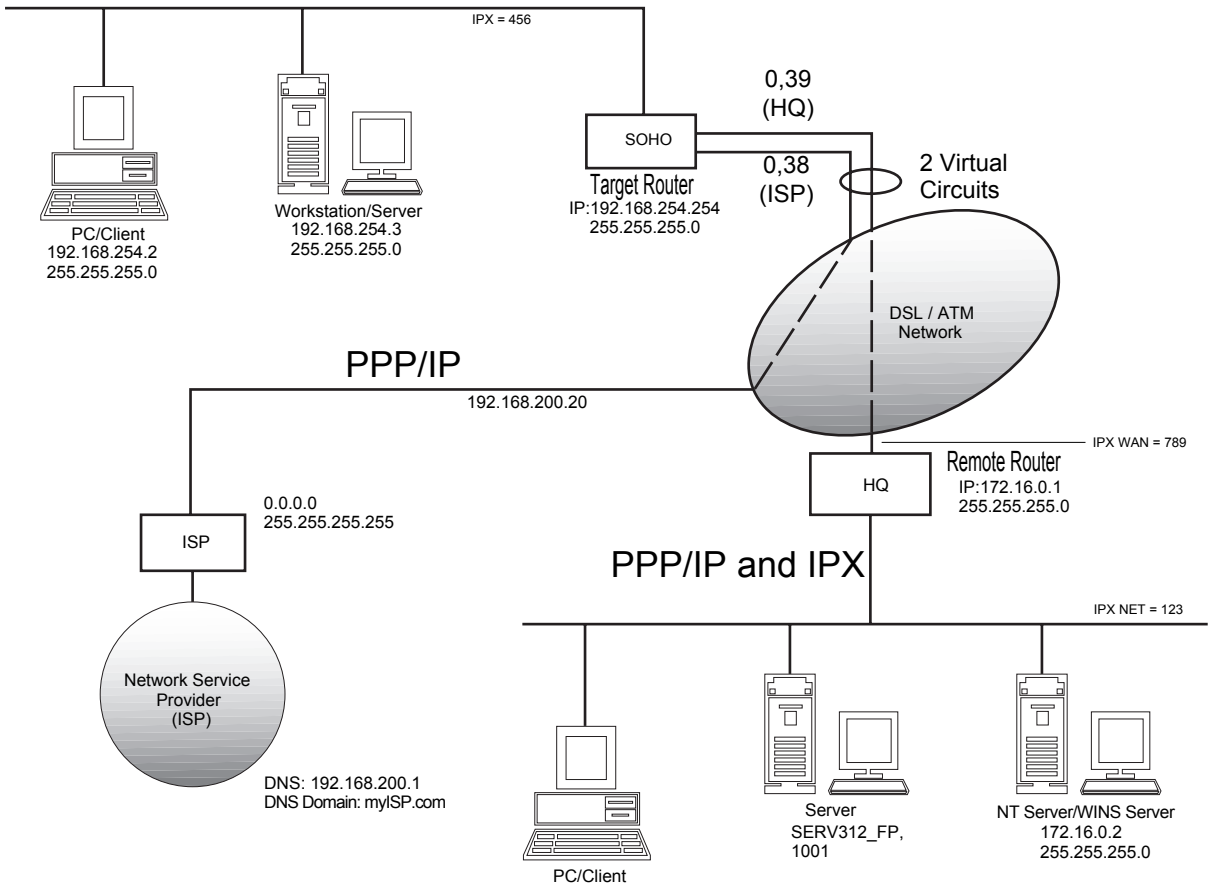
In this configuration example of a hypothetical network, a small office/home office (SOHO) will access:

- The Internet through an Internet Service Provider (ISP); it uses PPP as the link protocol with IP routing as the network protocol. Network Address Translation (NAT) is enabled to the ISP, because the ISP assigned the SOHO only one IP address.
- A central site (HQ) through a Network Service Provider (NSP provides access to the DSL/ATM Wide Area Network); it uses PPP as the Link Protocol with IP and IPX as its network protocols.

IP addresses are issued by the DHCP server. DHCP will be set up to issue DNS information to the SOHO LAN.

## Sample Configuration 1: Diagram for Target Router (SOHO)

Small Home Office SOHO (Target/Local Router)




Network Service Provider
(HQ)

## Sample Configuration 1: Tables for Target Router (SOHO)

| SOHO System Settings | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **System Settings** | | |
| Name | System Name | **system name SOHO** |
| Message | Message (optional) | **system msg Configured_Dec_1998** |
| Authentication Password | Authentication Password | **system password SOHOpasswd** |
| Ethernet IP Address | Ethernet IP Address and Subnet Mask (default IP address) | **eth ip addr 192.168.254.254 255.255.255.0** |
| Ethernet IPX Network | Ethernet IPX Network Number | **eth ipx addr 456** |
| **DHCP Settings** | | |
| DHCP Settings | DNS Domain Name | **dhcp set valueoption domainname myISP.com** |
| | DNS Server | **dhcp set valueoption domainnameserver 192.168.200.1** |
| | WINS Server Address | **dhcp set valueoption winsserver 172.16.0.2** |

| SOHO Remote Router Database<br>Entry: HQ | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **Remote Routers** | | |
| New Entry | Remote Router's Name | **remote add HQ** |
| Link Protocol | Link Protocol | **remote setProtocol PPP HQ** |
| PVC | VPI Number/VCI Number | **remote setPVC 0*39 HQ** |
| Security | Minimum Authentication (PAP is the default) | **remote setauthen PAP HQ** |
| | Remote Router's Password | **remote setpasswd HQpasswd HQ** |
| Bridging | Bridging on/off (Bridging is off by default) | **remote disbridge HQ** |
| TCP/IP Route Addresses | Remote Network's IP Addresses, Subnet Masks, and Metric | **remote addiproute 172.16.0.0 255.255.255.0 1 HQ** |
| IPX Address | Network #, Hop Count, Ticks | **remote addipxroute 1001 1 4 HQ** |
| IPX SAPs | SAPS: Server Name, Server Type, Network #, Node #, Sockets, type, hops | **remote addipxsap SERV312_FP 4 1001 00-00-00-00-00-01 451 3 1 HQ** |
| | WAN Network # | **remote setipxaddr 789 HQ** |

**Note:** Fill in one worksheet for each remote router in the remote router database.

| SOHO Remote Router Database<br>Entry: ISP | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **Remote Routers** | | |
| New Entry | Remote Router's Name | **remote add ISP** |
| Link Protocol | Link Protocol | **remote setProtocol PPP ISP** |
| PVC | VPI Number/VCI Number | **remote setPVC 0*38 ISP** |
| Security | Minimum Authentication<br>(PAP is the default)<br>Remote Router's Password | **remote setauthen PAP ISP**<br>**remote setpasswd ISPpasswd ISP** |
| Bridging | Bridging on/off<br>(Bridging is off by default) | **remote disbridge ISP** |
| TCP/IP Route Addresses | Remote Network's IP Addresses, Subnet Masks, and Metric<br>Network Address Translation<br>In Advanced:<br>Source WAN IP Address and Subnet Mask[a] | **remote addiproute 0.0.0.0 255.255.255.255 1 ISP (Default Route)**<br>**remote setiptranslate on ISP**<br>**remote setsrcipaddr 192.168.200.20 255.255.255 255 ISP** |

a This is needed only if the ISP does not assign an IP address automatically.
**Note:** Fill in one worksheet for each remote router in the remote router database

| SOHO Routing controls | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **IP and IPX Routing** | | |
| | TCP/IP Routing on/off | **eth ip enable** |
| | IPX Routing on/off | **eth ipx enable** |
| | Internet Firewall on/off<br>(Firewall is on by default) | **eth ip firewall on** |

## Sample Configuration 1: Check the Configuration with the LIST Commands

Type the following **commands** to obtain a list of your configuration.

**system list**

```
GENERAL INFORMATION FOR <SOHO>
  System started on..................... 12/1/1998 at 17:41
  Authentication override.............. NONE
 WAN to WAN Forwarding................. yes
  BOOTP/DHCP Server address............ none
  Telnet Port.......................... default (23)
SNMP Port.............................. default (161)
  System message: configured Dec-1998
```

**remote list**

```
INFORMATION FOR <HQ>
  Status............................... enabled
  Protocol in use...................... PPP
  Authentication....................... enabled
  Authentication level required........ PAP
  Connection Identifier (VPI*VCI)...... 0*39
  IP address translation............... off
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           172.16.0.0/255.255.255.0/1
  IPX network number................... 00000789
  Total IPX remote routes.............. 1
           00001001/1/4
  Total IPX SAPs....................... 1
    SERV312_FP 00001001 00:00:00:00:00:01 0451 0003 1
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes

INFORMATION FOR <ISP>
  Status............................... enabled
  Protocol in use...................... PPP
  Authentication....................... enabled
  Authentication level required........ PAP
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation............... on
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 192.168.200.20/255.255.255.255
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           0.0.0.0/255.255.255.255/1
  IPX network number................... 00000000
```

```
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
```

**dhcp list**

```
       bootp server ................. none
       bootp file ................... n/a

       DOMAINNAMESERVER (6) ......... 192.168.200.1
       DOMAINNAME (15) .............. myISP.com
       WINSSERVER (44) .............. 172.16.0.2

Subnet 192.168.254.0, disabled - other DHCP servers detected
       When DHCP servers are active . stop
       Mask ......................... 255.255.255.0
       first ip address ............. 192.168.254.2
       last ip address .............. 192.168.254.20
       lease ........................ default
       bootp ........................ not allowed
       bootp server ................. none
       bootp file ................... n/a
```

**eth list**

```
ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:A1:BF
  Bridging enabled..................... no
  IP Routing enabled................... yes
    Firewall filter enabled ........... yes
    Send IP RIP to the LAN............. rip-1 compatible
      Advertise me as default router... yes
    Process IP RIP packets received.... rip-1 compatible
      Receive default route by RIP..... yes
  RIP Multicast address................ default
  IP address/subnet mask............... 192.168.254.254/255.255.255.0
  IP static default gateway............ none
  IPX Routing enabled.................. yes
    External network number............ 00000456
    Frame type......................... 802.2
```

## Information About Names and Passwords for Sample Configuration 1

In this configuration example, the PPP Link Protocol requires using system names and passwords.

- ### System Passwords

  **SOHO** has a system password “SOHOpasswd,” which is used when SOHO communicates with HQ for authentication by that site and at any time when HQ challenges SOHO.

  **HQ** has a system password “HQpasswd,” which is, likewise, used when HQ communicates with site SOHO for authentication by SOHO and at any time SOHO challenges HQ.
  **ISP** has a system password “ISPpasswd” used for the same purpose.

- ### Remote Passwords

  Each router has a remote router’s password for each remote router defined in its Remote Router Database. The router will use the remote password to authenticate the remote router when the remote router communicates with or is challenged by the local site.

  For example, SOHO has remote router entries for HQ and ISP; defined in each table entry is the respective remote router’s password.

  The following table shows the names and passwords for each router that must be defined for authentication to be performed correctly. (This assumes that all three systems use some form of authentication protocol.)

  **Note:** If you experience trouble with passwords, we recommend that you set the remote router security to **disable authentication** to simplify the process.

| | **Names & Passwords Configured in SOHO Router** | **Names & Passwords Configured in HQ Router** | **Names & Passwords Configured in ISP Router** |
|---|---|---|---|
| **System Name** | SOHO | HQ | ISP |
| **System Password** | SOHOpasswd | HQpasswd | ISPpasswd |
| **Remote Router Database** | HQpasswd<br>ISPpasswd | SOHOpasswd | SOHOpasswd |

## Sample Configuration 2: RFC 1483 with IP and Bridging

This configuration example comprises:

- A scenario describing this configuration of the router SOHO
- A diagram showing the configuration information needed for this example
- Tables containing the configuration settings for this example
- Several **list** command outputs that are used to check the information entered for this particular configuration

**Note 1:** Names and passwords are *not* required with the RFC 1483 Link Protocol.

**Note 2:** Blank Network Information Worksheets are available to fill in the information for your own configuration in Appendix A.

### Scenario:

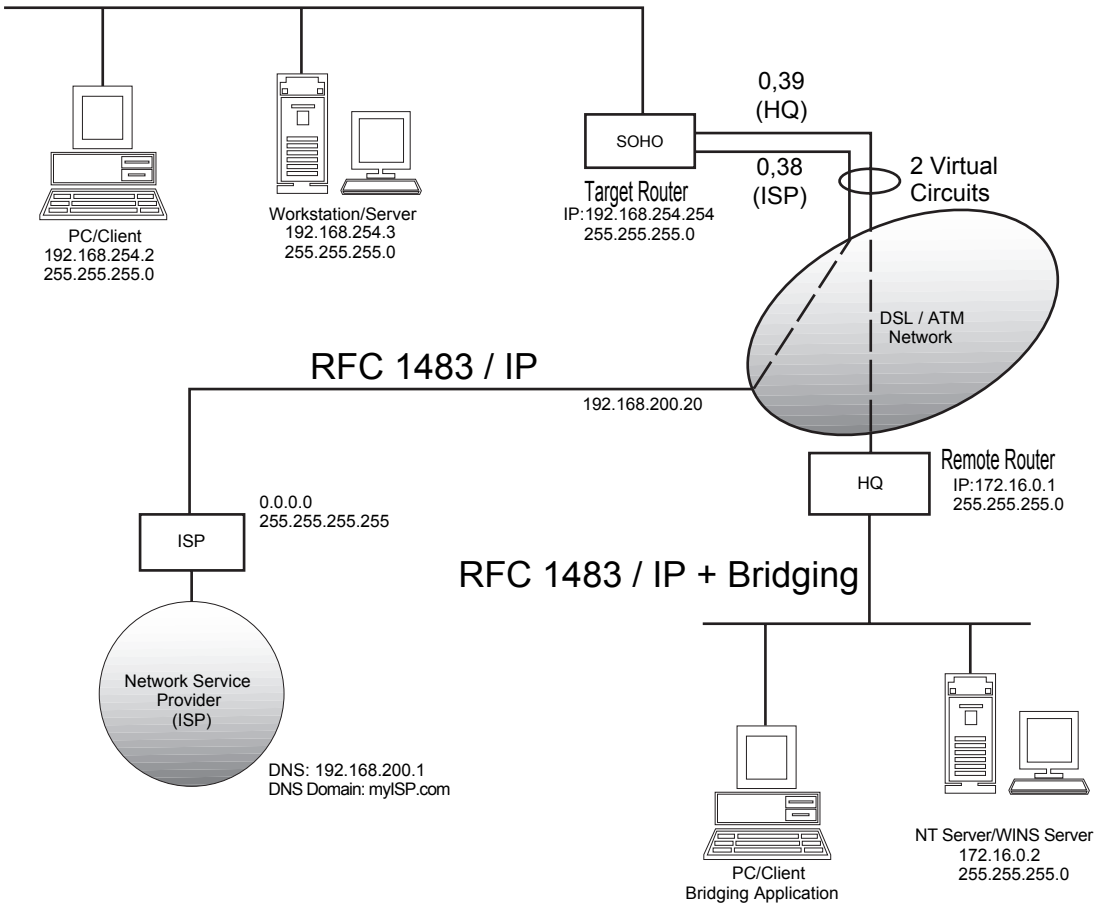
In this configuration example of a hypothetical network, a small office/home office (SOHO) will access:

- The Internet through an Internet Service Provider (ISP); it uses RFC 1483 as the Link Protocol with IP routing as the network protocol. Network Address Translation (NAT) is enabled to the ISP, since the ISP assigned SOHO only one IP address.
- A central site (HQ) through a Network Service Provider (NSP provides access to the DSL/ATM Wide Area Network); it uses RFC 1483 as the link protocol with bridging and IP routing as its network protocols.

IP addresses are issued by the DHCP server. DHCP will be set up to issue DNS information to the SOHO LAN.

## Sample Configuration 2: Diagram for Target Router SOHO

Small Home Office SOHO (Target Router)




**Network Service Provider (HQ)**

## Sample Configuration 2 : Tables for Target Router (SOHO)

| SOHO System Settings | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **System Settings** | | |
| Message | Message (optional) | **system msg RFC1483_dec98** |
| Ethernet IP Address | Ethernet IP Address and Subnet Mask<br>(default IP address) | **eth ip addr 192.168.254.254 255.255.255.0** |
| DHCP Settings | DNS Domain Name | **dhcp set valueoption domainname myISP.com** |
| | DNS Server | **dhcp set valueoption domainnameserver 192.168.200.1** |
| | WINS Server address | **dhcp set valueoption winsserver 172.16.0.2** |

| SOHO Remote Router Database<br>Entry: HQ | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **Remote Routers** | | |
| New Entry | Remote Router's Name | **remote add HQ** |
| Link Protocol | Link Protocol | **remote setProtocol RFC1483 HQ** |
| PVC | VPI Number/VCI Number | **remote setPVC 0*39 HQ** |
| Bridging | Bridging on/off | **remote enabridge HQ** |
| TCP/IP Route Addresses | Remote Network's IP Addresses, Subnet Masks, and Metric | **remote addiproute 172.16.0.0 255.255.255.0 1 HQ** |

| SOHO Remote Router Database<br>Entry: ISP | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **Remote Routers** | | |
| New Entry | Remote Router's Name | **remote add ISP** |
| Link Protocol | Link Protocol | **remote setProtocol RFC1483 ISP** |
| PVC | VPI Number/VCI Number | **remote setPVC 0*38 ISP** |
| Bridging | Bridging On/Off<br>(Bridging is Off by default) | **remote disbridge ISP** |
| TCP/IP Route Addresses | Remote Network's IP Addresses, Subnet Masks, and Metric | **remote addiproute 0.0.0.0 255.255.255.255 1 ISP** |
| | Network Address Translation (NAT) | **remote setiptranslate on ISP** |
| | In Advanced: | |
| | Source WAN IP Address and Subnet Mask | **remote setsrcipaddr 192.168.200.20 255.255.255.255 ISP** |

| SOHO Routing Controls | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **IP and IPX Routing** | | |
| **IP and IPX Routing** | TCP/IP Routingon/off | **eth ip enable** |
| | IPX Routing on/off<br>(IPX routing is off by default) | **eth ipx disable** |
| | Internet Firewall on/off<br>(Firewall is on by default) | **eth ip firewall on** |

## Sample Configuration 2: Check the Configuration with the LIST Commands

**system list**

```
GENERAL INFORMATION FOR <SOHO>
  System started on..................... 12/1/1998 at 17:48
  Authentication override.............. NONE
WAN to WAN Forwarding.................. yes
  BOOTP/DHCP Server address............ none
  Telnet Port.......................... default (23)
SNMP Port.............................. default (161)
  System message: ADSL RFC1483 sample
```

**eth list**

```
ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:A1:BF
  Bridging enabled..................... yes
  IP Routing enabled................... yes
    Firewall filter enabled ........... yes
    Send IP RIP to the LAN............. rip-1 compatible
      Advertise me as default router... yes
    Process IP RIP packets received.... rip-1 compatible
      Receive default route by RIP..... yes
  RIP Multicast address................ default
  IP address/subnet mask............... 192.168.254.254/255.255.255.0
    IP static default gateway.......... none
  IPX Routing enabled.................. no
    External network number............ 00000000
    Frame type.......................... 802.2
```

**remote list**

```
INFORMATION FOR <HQ>
  Status............................... enabled
  Protocol in use...................... RFC1483 (SNAP)
  Connection Identifier (VPI*VCI)...... 0*39
  IP address translation............... off
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
  Receive IP RIP from this dest........ no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           172.16.0.0/255.255.255.0/1
  IPX network number................... 00000000
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... yes
    Exchange spanning tree with dest... yes

INFORMATION FOR <ISP>
  Status............................... enabled
  Protocol in use...................... RFC1483 (SNAP)
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation............... on
```

```
  Compression Negotiation.............. off
  Source IP address/subnet mask........ 192.168.200.20/255.255.255.255
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no

Receive IP RIP from this dest......... .no
    Receive IP default route by RIP.... no
  Keep this IP destination private..... yes
  Total IP remote routes............... 1
           0.0.0.0/255.255.255.255/1
  IPX network number................... 00000000
  Total IPX remote routes.............. 0
  Total IPX SAPs....................... 0
  Bridging enabled..................... no
    Exchange spanning tree with dest.... yes
```

**dhcp list**

```
       bootp server ................. none
       bootp file ................... n/a

       DOMAINNAMESERVER (6) ......... 192.168.200.1
       DOMAINNAME (15) .............. myISP.com
       WINSSERVER (44) .............. 172.16.0.2

Subnet 192.168.254.0, disabled - other DHCP servers detected
       When DHCP servers are active . stop
       Mask ......................... 255.255.255.0
       first ip address ............. 192.168.254.2
       last ip address .............. 192.168.254.20
       lease ........................ default
       bootp ........................ not allowed
       bootp server ................. none
       bootp file ................... n/a
```

## Sample Configuration 3: Configuring a Dual-Ethernet Router for IP Routing

### Scenario:

The following example provides a simple sample configuration for a Dual-Ethernet router (eth_router) with IP routing enabled.

The router's hub (ETH/0) belongs to the 192.168.254.0 subnet. The router's ETH/1 belongs to the 192.168.253.0 subnet.

ETH/0 will route packets to ETH/1 at the address 192.168.253.254. DHCP is enabled for both subnets.

| **eth_router Configuration** | | |
|---|---|---|
| **Configuration Section** | **Item** | **Commands** |
| **System Settings** | | |
| Name | System Name (optional) | **system name eth_router** |
| Message | Message (optional) | **system msg Configured_Jan_1999** |
| **Ethernet Settings** | | |
| Routing/ Bridging Controls | Enable IP routing | **eth ip enable** |
| | Disable bridging | **eth br disable** |
| ETH/0 IP Address | Define ETH/0 IP address for the hub side | **eth ip addr 192.168.254.254 255.255.255.0 0** |
| ETH/1 IP Address | Define ETH/1 IP address for the single 10Base-T side | **eth ip addr 192.168.253.254 255.255.255.0 1** |
| TCP/IP default route address | ETH/0 sends all traffic to ETH/1 | **eth ip addroute 0.0.0.0 255.255.255.255 192.168.253.254 1 1** |
| **DHCP Settings** | | |
| DHCP Settings | Define DHCP network for ETH/1 | **dhcp add 192.168.253.0 255.255.255.0** |
| | Create an address pool for ETH/1 | **dhcp set addresses 192.168.253.2 192.168.253.20** |
| | DNS Domain Name | **dhcp set valueoption domainname myISP.com** |
| | DNS Server | **dhcp set valueoption domainnameserver 192.168.200.1** |
| | WINS Server Address | **dhcp set valueoption winsserver 172.16.0.2** |

# Chapter 4. Configuring Special Features

The features described in this chapter are advanced topics. They are primarily intended for experienced users and network administrators to perform network management and more complex configurations.

- Bridge Filtering and IP firewall
- IP protocol controls (RIP)
- Dynamic Host Configuration Protocol  (DHCP)
- Network Address Translation (NAT )
- Management security

The folowing features can be purchased as software option keys. To determine which software options are installed on your router, use the **vers** command. (If a feature has not been enabled, it is listed with a ~ prefix.)

- Encryption
- IP filtering
- L2TP tunneling
- IPSec (Internet Protocol Security) and IKE (Internet Key Exchange)
- 3DES encryption

# Multiple IP Subnets

You may configure the router to provide access to multiple IP subnets on the Ethernet network. (This feature does not apply to IPX or bridged traffic.)

Each IP subnet is referenced as a logical Ethernet interface. You may define multiple logical interfaces for each physical Ethernet interface (that is, port) in the router. Each logical interface is referenced by its port number and logical interface number (*port #:logical#)*.

To define a logical interface to route traffic for an IP subnet, first use the **eth add** command (page 167); it specifies the port number and the new logical interface number. You then enter an **eth ip addr** command (page 168) to define the IP address and subnet mask of the IP subnet.

The default logical interface for each port is interface 0; this logical interface 0 always exists and cannot be deleted. (Other logical interfaces may be deleted using the **eth delete** command [page 168].)

After the **eth add** and **eth ip addr** commands define the Ethernet logical interface, the logical interface can be referenced by other **eth** commands, including:

- **eth ip addRoute** Adds an Ethernet IP route that uses the logical Ethernet interface. The route is added to the default routing table (page 169).
- **eth ip bindRoute** Adds an Ethernet IP route that uses the logical Ethernet interface. The route is added to a virtual routing table (page 169).
- **eth ip filter** Manages IP filters for the logical Ethernet interface. Lists of input, output, and forward filters may be defined for the interface (page 172).
- **eth ip options** Sets RIP options for the logical interface; these options set IP routing information protocol controls (page 178).

**Note:** In general, these commands are not effective until you enter **save** and **reboot**. However, the **eth ip bindRoute** and **eth ip filter** commands will take effect before a **save** and **reboot** if the logical Ethernet interface is already active.

# Virtual Routing Tables

The virtual routing feature allows you to define multiple routing tables. This is also known as IP virtual router support.

To define a new routing table, you must specify a name for the routing table and a range of IP source addresses that use that table. The router determines which routing table to use based on the source address in the packet. For example, if the router receives a packet whose source address is 192.168.254.10, it checks if that address is within the address range defined for a virtual routing table. If it is, the virtual routing table is used to route the packet. If it is not, the default routing table is used instead.

The address ranges assigned to the virtual routing tables may not overlap. All source IP addresses not assigned to a virtual routing table are routed using the default routing table. You can add routes to the default routing table using **eth ip addroute** (page 169) and **remote addiproute** (page 183).

The following commands define the names and address ranges of the virtual routing tables:

**system addIPRoutingTable** Adds a range of IP addresses to a virtual routing table. The virtual routing table is defined if it does not already exist (page 154).

**system delIPRoutingTable** Deletes a range of IP addresses from the range defined for a virtual routing table or deletes the entire table (page 159).

**system moveIPRoutingTable** Moves a range of IP addresses from their current assignment to the specified virtual routing table. The virtual routing table is defined if it does not already exist (page 163).

To add and remove routes from a virtual routing table, use these commands:

**eth ip bindRoute** Adds an Ethernet route to a virtual routing table (page 169).

**eth ip unbindRoute** Removes an Ethernet route from a virtual routing table (page 179).

**remote bindIPVirtualRoute** Adds a remote route to a virtual routing table (page 186).

**remote unbindIPVirtualRoute** Removes a remote route from a virtual routing table (page 207).

**Note:** Unlike changes to the default routing table, changes to IP virtual routing tables take effect immediately. However, the changes are lost if they are not saved before the next **reboot**.

# Bridge Filtering and IP Firewall

You can control the flow of packets across the router using bridge filtering. Bridge filtering lets you “deny” or “allow” packets to cross the network based on position and hexadecimal content within the packet. This enables you to restrict or forward messages with a specified address, protocol, or data content. Common uses are to prevent access to remote networks, control unauthorized access to the local network, and limit unnecessary traffic.

For example, it might be necessary to restrict remote access for specific users on the local network. In this case, bridging filters are defined using the local MAC address for each user to be restricted. Each bridging filter is specified as a "deny" filter based on the MAC address and position of the address within the packet. To initiate bridge filtering, “deny” filtering mode is then enabled. Every packet with one of the MAC addresses would not be bridged across the router until “deny” filtering mode was disabled.

Similarly, protocol filtering can be used to prevent a specific protocol from being bridged. In this case, the protocol id field in a packet is used to deny or allow a packet. You can also restrict, for example, the bridging of specific broadcast packets.

## Configure Bridge Filtering

Bridge filtering allows you to control the packets transferred across the router. This feature can be used to enhance security or improve performance. The filtering is based on matched patterns within the packet at a specified offset. Two filtering modes are available:

- “Deny” mode will discard any packet matched to the “deny” filters in the filter database and let all other packets pass.
- “Allow” mode will only pass the packets that match the “allow” filters in the filter database and discard all others.

Up to 40 “allow” filters or 40 “deny” filters can be activated from the filter database.

Enter the filters, including the pattern, offset, and filter mode, into a filter database. If you intend to restrict specific stations or subnetworks from bridging, then add the filters with a “deny” designation and then enable “deny” filtering. If you wish to allow only specific stations or subnetworks to bridge, then add the filters with an “allow” designation and enable “allow” filtering. Add each filter with the following command:

**filter br add** [*pos*] [*data*] [deny | allow]

where [*pos*] is the byte offset within a packet (number from 0-127) to a [*data*] (a hex number up to 6 bytes). This data and offset number can be used to identify an address, a protocol id, or data content. After entering your filters, verify your entries with the following command:

**filter br list**

If you have entered an incorrect filter, delete the filter using the **filter br del** command. When you are satisfied with the filter list, save the filtering database with the **save filter** command. You must reboot the router to load the filtering database. Then enable bridging filtering with the following command:

**filter br use** [none | deny | allow]

To test the filtering configuration, access the remote destination identified in the filter.

## Enable/Disable Internet Firewall Filtering

The router supports IP Internet Firewall Filtering to prevent unauthorized access to your system and network resources from the Internet. This filter discards packets received from the WAN that have a source IP address recognized as a local LAN address.

**Caution:** This is a simple firewall check; it does not add much security. For more elaborate firewall features, see IP Filtering, page 103.

You can control the Internet Firewall Filtering feature using the command:

**eth ip firewall** [on | off | list]

The Internet Firewall defaults to on during initial configuration, but it is active *only* when Ethernet LAN IP routing is on.

As described earlier, Ethernet LAN IP routing is controlled by the commands:

**eth ip enable**
**eth ip disable**

Therefore, at initial configuration, to activate the Internet Firewall Filter, you need only enable IP routing. If you do not wish the router to perform IP Internet Firewall Filtering while doing IP routing, you must turn off the firewall filter. *Remember to save and reboot* if you alter the IP routing status.

# IP (RIP) Protocol Controls

You can configure the router to send and receive RIP packet information, respectively, to and from the remote router. This means that the local site will "learn" all about the routes beyond the remote router and the remote router will "learn" all about the local site's routes. You may not want this to occur in some cases. For example, if you are connecting to a site outside your company, such as the Internet, you may want to keep knowledge about your local site's routes private.

The default is to not send or receive IP RIP packets. If RIP packets are not allowed to flow on the WAN link, you *must* use the **remote addiproute** command to configure static routes for this WAN link. You can also advertise the local site's existence. The default is to keep the local site's existence private.

If you wish to allow sending or receiving RIP packets or default routes or to advertise the local site's existence, use the following command:

**remote setipoptions <***option***>** [**on**|**off**] <*remoteName*>

where <*option*> is:

| | |
|---|---|
| rxrip | Receive IP RIP packets from the remote destination |
| rxrip1 | Receive and process RIP-1 packets only |
| rxrip2 | Receive and process RIP-2 packet only |
| rxdef | Receive the remote site's default route |
| txrip | Send IP RIP packets to the remote destination |
| txrip1 | Send RIP-1 packets only |
| txrip2 | Send RIP-2 packets only |
| txdef | Send the local site's default route |
| private | Keep the local site's existence private |

**Note:** RIP can be set on the LAN interface as well. See the **eth ip** options commands for more information.

# DHCP (Dynamic Host Configuration Protocol)

This section describes how to configure DHCP using the Command Line Interface. Configuring DHCP can be a complex process; this section is therefore intended for network managers. For a complete list and explanation of the DHCP commands, see DHCP (Dynamic Host Configuration Protocol) Commands, page 227.

**Note:** Some DHCP values can be set using the Windows Quick Start application, the Windows Configuration Manager, or the web-based EZ Setup application.

## DHCP Address Allocation

The router supports DHCP and acts as the DHCP server. DHCP is a service that allocates IP addresses *automatically* to any DHCP client requesting an IP address. (A DHCP client can be any device attached to your network, for example, a PC.) It can also provide option values (such as the subnet mask, DNS, and gateway values) automatically.

Using DHCP to automatically acquire initialization parameters translates into avoiding the more involved router/PC manual initialization process. (The manual initialization requires reconfiguration of router and/or PC addresses to be in the same network.)

To configure DHCP for a network, the network administrator defines a range of valid IP addresses to be used in the subnetwork as well as options and other parameters. This process is described in the next section, *DHCP Administration and Configuration*.

**Note 1:** DHCP will work only if the TCP/IP stack is installed on the PCs.

**Note 2:** In Windows, DHCP is enabled by selecting it on your PC (under **Settings, Control Panel, Network,** and **TCP/IP** in the **Configuration** tab page).

## DHCP Client Requests

Before becoming active, the router's DHCP server attempts to locate other active DHCP servers on the network, such as Windows NT servers. If one is detected, the router's DHCP server disables itself.

When the WAN link activates and the source IP address or mask is undefined (i.e. 0.0.0.0), the router places a DHCP client request over the WAN link. The router may learn the following parameters:

- DNS address
- Default gateway
- Syslog server IP address
- Time server IP address
- Source IP address to use

To see the gateway and source IP addresses that were returned, use the **iproutes** command.

The IP addresses and options assigned to a client are collectively called the "lease". The lease is only valid for a certain period of time and is automatically renewed by the client.

# DHCP Administration and Configuration

The DHCP administration and configuration process is divided into the following parts:

- Manipulating subnetworks and explicit client leases
- Setting option values
- Managing BootP
- Defining option types
- Configuring BootP/DHCP relays
- Other information

**Note:** To save the DHCP configuration or changes to flash memory in the router, remember to use the command **dhcp save**.

## Manipulating Subnetworks and Explicit Client Leases

### Enabling/Disabling a Subnetwork or a Client Lease

To enable/disable a subnetwork or a client lease, use the commands:

**dhcp enable** all | **<***net***> <***ipaddr***>**
**dhcp disable** all | **<***net***> <***ipaddr***>**

**`Examples:`**

To enable the subnetwork 192.168.254.0 if that subnetwork exists, enter:

```
dhcp enable 192.168.254.0
```

To enable the client lease 192.168.254.17 if that client lease exists, enter:

```
dhcp enable 192.168.254.17
```

To disable the client lease 192.168.254.18 if that client lease exists, enter:

```
dhcp disable 192.168.254.18
```

To check the results of these commands, use: **dhcp list**

If the client lease does not exist, it must be explicitly created.

### Adding Subnetworks and Client Leases

- #### Adding a Subnetwork

  The following commands are used to add/delete subnetworks. Only *one* subnetwork with *one* pool of IP addresses may be defined for a subnet.

  To add a subnetwork, use:

**dhcp add <*net*> <*mask*>**

To remove a subnetwork, use:

**dhcp del <*net*>**

**Note:** All client leases associated with this subnetwork are automatically deleted.

**Example 1:**

The following command creates a subnetwork 192.168.254.0 with a subnet mask of 255.255.255.0:

```
dhcp add 192.168.254.0 255.255.255.0
```

**Example 2:**

The following command deletes the subnetwork 192.168.254.0 *and* deletes *all* client leases associated with that subnetwork:

```
dhcp del 192.168.254.0
```

- ## Adding Explicit or Dynamic Client Leases

Client leases may either be created dynamically or explicitly. Usually client leases are created dynamically when PCs boot and ask for IP addresses.

### Explicit client leases

To add an explicit client lease, a subnetwork *must* already exist (use **dhcp add <***net***> <***mask***>** to add the subnetwork) before the client lease may be added. Use the command:
**dhcp add <***ipaddr***>**

To remove a client lease, use:
**dhcp del <***ipaddr***>**

**Note:** An administrator *may* create a client lease that is part of a subnet but does not fall within the pool of IP addresses.

**Example 1:**

To explicitly add the client lease 192.168.254.31, type:
**`dhcp add 192.168.254.31`**

**Example 2:**

To delete the client lease 192.168.254.31, type:
d**`hcp del 192.168.254.31`**

### Dynamic Client Leases

Dynamic client leases are created from the pool of IP addresses associated with that subnetwork.

To set or change the pool, use:
**dhcp set addresses <***first ip addr***> <***last ip addr***>**

To clear the values from the pool, use:
**dhcp clear addresses <***net***>**

**Note:** Any client leases that currently exist will not be affected.

To remove a client lease that was dynamically created, use:
**dhcp del <***ipaddr***>**

**Caution:** If <*ipaddr*> is a subnet, you will delete the entire subnet.

## Setting the Lease Time

- ## Concepts

The information given by the DHCP server (router) to your PC is leased for a specific amount of time. The client lease has already been selected. The DHCP server will select the lease time based on the option defined for the client lease as described by this algorithm:

1. If the client lease option is a specific number or is infinite, then the server uses the specified lease time associated with this client lease.

2. If the client lease option is “default”, then the server goes up one level (to the subnetwork) and uses the lease time explicitly specified for the subnetwork.

3. If the client *and* subnetwork lease options are both “default”, then the server goes up one level (global) and uses the lease time defined at the global level (server).

4. Lease time:
   The minimum lease time is 1 hour.
   The global default is 168 hours.

- ### Commands

The following commands are used by network administrators to control lease time.

To set the lease time explicitly for the client lease, use:
**dhcp set lease <***ipaddr***> <***hours***>**

To set the lease time explicitly for the subnetwork lease, use:
**dhcp set lease <***net***> <***hours***>**

To set the lease time explicitly for the global lease, use:
**dhcp set lease <***hours***>**

**Example 1:**

To set the lease time to “default” for the client 192.168.254.17, type:
```
dhcp set lease 192.168.254.17 default
```

**Example 2:**

To set the subnetwork lease time to infinite for the subnet 192.168.254.0, type:
```
dhcp set lease 192.168.254.0 infinite
```

**Example 3:**

To set the global lease time to 2 hours, type:
```
dhcp set lease 2
```

## Manually Changing Client Leases

In general, administrators do not need to change client leases manually. However, if the need arises to do so, the following two commands are used.

**Warning**: The client will not be aware that the administrator has changed or released a client lease!

To change the client lease expiration time to a given value:
**dhcp set expire <***ipaddr***> <***hours***>**

Setting the expiration time to “default” will cause the server to compute the lease time using the algorithm as described in Setting the Lease Time, page 84.

To release the client lease so it becomes available for other assignments:
**dhcp clear expire <***ipaddr***>**

# Setting Option Values

Administrators can set values for global options, for options specific to a subnetwork, or for options specific to a client lease.

**Note:** See RFC 2131/2132 for the description of various options.

## Concepts

The server returns values for options explicitly requested in the client request. It selects the values to return based on the following algorithm:

1. If the value is defined for the client, then the server returns the requested value for an option.
2. If the value for the option has not been set for the client, then the server returns the value option if it has been defined for the subnetwork.
3. If the value option does not exist for the client *and* does not exist for the subnetwork, then the server returns the value option if it has been defined globally.
4. If the value option is not defined anywhere, the server does *not* return any value for that option in its reply to the client request.

**Important:** When the server replies to a client:

- *It does not* return any option values *not* requested by the client.
- *It does not* support the definition of a "class" of clients.
- *It does not* return any non-default option values *unless* the client requests the option value *and* the server has a value defined for that option.
- *It does not* return any non-default values on the clients subnet *unless* the client requests the value for that option.

## Commands for Global Option Values

To set the value for a global option, use:
**dhcp set valueoption <*code*> <*value*>...**

The code can be a number between 1 and 61 or a keyword.

To see the list of predefined and user-defined options, use:
**dhcp list definedoptions**

To clear the value for a global option, use:
**dhcp clear valueoption <*code*>**

**Example:**

To set the global value for the domain name server option, enter:

```
dhcp set valueoption domainnameserver 192.168.254.2 192.168.254.3
```

## Commands for Specific Option Values for a Subnetwork

To set the value for an option associated with a subnetwork, use:
**dhcp set valueoption <*net*> <*code*> <*value*>...**

To clear the value for an option associated with a subnetwork, use:
**dhcp clear valueoption <*net*> <*code*>**

**Examples:**

```
dhcp set valueoption 192.168.254.0 gateway 192.168.254.254
dhcp set valueoption 6 192.84.210.75 192.84.210.68
```

### Commands for Specific Option Values for a Client Lease

To set the value for an option associated with a specific client, use:
**dhcp set valueoption** <*ipaddr*> **<***code***>** **<***value***>...**

To clear the value for an option associated with a specific client, use:
**dhcp clear valueoption <***ipaddr***>** **<***code***>**

**Example:**

```
dhcp set valueoption 192.168.254.251 winserver 192.168.254.7
```

### Commands for Listing and Checking Option Values

To list the values for global options as well as subnet and client lease information, use:
**dhcp list**

To list options that are set for that subnet/client lease as well as subnet/client lease information, use:
**dhcp list** <*net*>|<*ipaddr*>

This command lists all available options (predefined and user-defined options):
**dhcp list definedoptions**

This command lists all available options starting with the string “name”.
**dhcp list definedoptions name**

To list the lease time use:
**dhcp list lease**

**Example:**

This command lists the subnet 192.168.254.0 including any options set specifically for that subnet:

```
dhcp list 192.168.254.0
```

## Managing BootP

Administrators can enable and disable BootP and specify the BootP server. BootP can be enabled at the subnetwork and at the client lease level.

**Note:** By default, the DHCP server does *not* satisfy BootP requests unless the administrator has explicitly enabled BootP (at the subnetwork or lease level).

### About BootP and DHCP

BootP and DHCP provide services that are very similar. However, as an older service, BootP offers only a subset of the services provided by DHCP.

The main difference between BootP and DHCP is that the client lease expiration for a BootP client is always *infinite*.

**Note:** Remember, when BootP is enabled, the client assumes that the lease is infinite.

### Enable/Disable BootP

To allow BootP request processing for a particular client/subnet, use the command:
**dhcp bootp allow <***net***>|<***ipaddr***>**

To disallow BootP request processing for a particular client/subnet, type:
**dhcp bootp disallow** <*net*>|<*ipaddr*>

### Use BootP to Specify the Boot Server

The following commands let the administrator specify the TFTP server (boot server) and boot file name. The administrator should first configure the IP address of the TFTP server and file name (kernel) from which to boot. This is particularly useful if the kernel in the router's flash is corrupt or does not exist.

To set the IP address of the server and the file to boot from, use the commands:
**dhcp bootp tftpserver** [**<***net***>|<***ipaddr***>**] **<***tftpserver ipaddr***>**
**dhcp bootp file** [**<***net***>|<***ipaddr***>**] **<***file name***>**

To clear the IP address of the server and the file to boot from, use:
**dhcp bootp tftpserver** [<net>|<*ipaddr*>] 0.0.0.0

**Example 1:**

To set the global BootP server IP address to 192.168.254.7:
```
dhcp bootp tftpserver 192.168.254.7
```

**Example 2:**

To set the subnet 192.168.254.0 server IP address to 192.168.254.8:
```
dhcp bootp tftpserver 192.168.254.0 192.168.254.8
```

**Example 3:**

To set the client 192.168.254.21 server IP address to 192.168.254.9
```
dhcp bootp tftpserver 192.168.254.21 192.168.254.9
```

**Example 4:**

To set the subnet 192.168.254.0 boot file to "kernel.100":
```
dhcp bootp file 192.168.254.0 kernel.100
```

**Example 5:**

To clear the global BootP server IP address and file name:
```
dhcp bootp tftpserver 0.0.0.0
```

**Example 6:**

To clear the subnet 192.168.254.0 server IP address and file name:
```
dhcp bootp tftpserver 192.168.254.0 0.0.0.0
```

## Defining Option Types

### Concepts

A DHCP option is a code, length, or value. An option also has a "type" (byte, word, long, longint, binary, IP address, string).

The subnet mask, router gateway, domain name, domain name servers, NetBios name servers are all DHCP options. Refer to RFC 1533 if you require more information.

Usually users will *not* need to define their own option types. The list of predefined option types based on RFC 1533 can be shown by typing **dhcp list definedoptions.**

### Commands

The following commands are available for adding/deleting option types:
**dhcp add** <*code*> <*min*> <*max*> <*type*>

To list option types that are currently defined, use:
**dhcp list definedoptions...**

To list the definitions for all known options, use:
**dhcp list definedoptions**

To get help information, use:
**dhcp list definedoptions?**

To list the definition for option 1, if option 1 is defined, type:|
```
dhcp list definedoptions 1
```

To list the definition for all options that are well-known AND have a name starting with “h”, type:
```
dhcp list definedoptions h
```

**Example:**

To define a new option with a code of 128, a minimum number of IP addresses of 1, a maximum number of IP addresses of 4, of type “IP address”, type:
```
dhcp add 128 1 4 ipAddress
```

This information implies that:

- Some DHCP client will know about the option with code 128.
- Option 128 allows IP addresses.
- The server can have a minimum of 1 IP address.
- The server can have up to 4 IP addresses.
- The administrator will still need to set the option value either globally, specific to a subnetwork, or specific to a client for the option to have any meaning.

To delete the definition of the option with code 128, type:

```
dhcp del 128
```

The values for this option that have been set globally, specific to a subnetwork, or specific to a client will *not* be removed. The administrator must remove those values explicitly. Well-known type option codes *cannot* be changed or deleted.

## Configuring BootP/DHCP Relays

BootP/DHCP Relays are used by system administrators when the DHCP configuration parameters are acquired from a BootP/DHCP server other than the router’s DHCP server.

This feature allows configuration information to be centrally controlled. Enabling a BootP/DHCP Relay disables DHCP on the router because, by definition, only one policy mechanism can be supported.

BootP/DHCP Relays are enabled and disabled using the command:
**system bootpserver**

## DHCP Information File

DHCP information is kept in the file DHCP.DAT, a self-contained file.

This file contains *all* DHCP information including:

- the option definitions
- the subnetworks that have been added
- the client lease information
- the option values that have been set

This file can be uploaded/downloaded from one router to another.

# Network Address Translation (NAT)

The router supports classic NAT (one NAT IP address assigned to one PC IP address) and a NAT technique known as masquerading (one single NAT IP address assigned to many PC IP addresses).

## General NAT Rules

1. IP routing must be enabled.
2. NAT can be run on a per-remote-router basis.
3. Any number of PCs on the LAN may be going to the same or different remote routers at the same time. In reality, the number of PCs on the LAN that can be supported is limited by how much memory the router consumes maintaining table information *and* by how many connections are currently active.
4. Some operations will *not* work. Specifically, services that place IP address/port information in the data *may not work* until the router examines their packets and figures out what information in the data needs to be changed. Remember that the router is remapping both IP addresses and ports.
5. When using NAT with a remote router, either the remote ISP *must* supply the IP address for NAT translation or the user *must* configure the IP address for NAT translation locally.
6. Any number of PCs on the LAN may have a connection to the same or different remote routers at the same time. In reality, the number of PCs on the LAN that can be supported is limited by the amount of memory consumed by the router to maintain table information *and* by the number of connections the router "thinks" are currently active. Theoretically, up to 64,000 active connections per protocol type—TCP/UDP—can be concurrently running, if the table space is available.

## Masquerading

With masquerading, multiple local (PC) IP addresses are mapped to a single global IP address. Many local (PCs) IP addresses are therefore hidden behind a single global IP address. The advantage of this type of NAT is that users only need one global IP address, but the entire local LAN can still access the Internet. This NAT technique requires not only remapping IP addresses but also TCP and UDP ports.

Each PC on the LAN side has an IP address and a mask. When the router connects to an ISP, the router appears to be a "host" with one IP address and mask. The IP address that the router uses to communicate with the ISP is obtained dynamically (with PPP/IPCP or DHCP) or is statically configured. When the PC connects to the ISP, the IP address and port used by the PC are remapped to the IP address assigned to the router. This remapping is done dynamically.

### Client Configuration

- #### Enable NAT

  To enable NAT, use the commands:
  **remote setIpTranslate on <***remoteName***>**
  **save**

  The **save** command makes the above changes persistent across boots; these changes turn NAT on when you are connected to this remote router.

- ### Obtain an IP Address for NAT

  The IP address (the IP address "known" by the remote ISP) used for this type of NAT can be assigned in two ways.

  The ISP dynamically assigns the IP address. Use the commands:
  **remote setSrcIpAddr** 0.0.0.0 0.0.0.0 <*remoteName*>
  **save**

  The IP address is assigned locally. Use the commands:
  **remote setSrcIpAddr** *ww.xx.yy.zz* 255.255.255.255 <*remoteName*>
  **save**

  **Note:** *ww.xx.yy.zz* is the IP address that the user on the local LAN assigns.

## Server Configuration

This section is intended for users and network administrators who wish to allow WAN access to a Web server, FTP server, SMTP server, etc., on their local LAN, while using NAT.

NAT needs a way to identify which local PC [local IP address(es)] should receive these server requests. The servers can be configured on a *per-remote-router* basis as well as *globally*.

- ### Remote Commands

  The following two commands are used to enable/disable a local IP address (on your LAN) as the server for a particular protocol for the remote router <*remoteName*>.

  **remote addServer** <*ipaddr*> |discard|me <*protocolid*> tcp|udp <*first port*> ftp|telnet|smtp|snmp|http [<*last port*>[<*first private port*>]] <*remoteName*>

  **remote delServer** <*ipaddr*> |discard|me <*protocolid*> tcp|udp <*first port*> ftp|telnet|smtp|snmp|http [<*last port*>[<*first private port*>]] <*remoteName*>

  where

  *first port:* this is the first or only port as seen by the remote end.

  *last port:* if specified, this is used with <*first port*> to specify a range of ports as seen by the remote end for the server on your LAN.

  *first private port:* if specified, this is a port remapping of the incoming request from the remote end.
  *first port* maps to *first private port*.
  *first port* + 1 maps to *first private port* + 1.

  *last port* maps to *first private port* + *last port* - *first port*

  *first port* through *last port* are the ports as seen by the remote end.
  *first private port* through *first private port* + *last port* - *first port* are the equivalent ports through which the server on your local LAN will receive the request.

  This command is used to view all of the remote entries, including the changes.
  **remote list <***remoteName***>**

  Remember to type **save** to make the changes persistent across boots.

**Example 1:**

Assume that the local LAN network is 192.168.1.0 255.255.255.0. The following commands are typed to enable a Telnet server on the local LAN with the IP address 192.168.1.3, and an FTP server with the IP address 192.168.1.2.

```
remote addServer 192.168.1.3 tcp telnet router1
remote addServer 192.168.1.2 tcp ftp router1
```

When the local router receives a request from *router1* to communicate with the local Telnet server, the local router will send the request to 192.168.1.3. If *router1* asks to talk to the local FTP server, the local router will send the request to 192.168.1.2.

**Example 2:**

Assume that the local LAN network is 192.168.1.0 255.255.255.0. When the port value of 0 (zero) is used, it directs all ports of the specified protocol to the IP address specified.

```
remote addServer 192.168.1.4 tcp 0 router1
```

**Note: addserver** commands using specific port numbers take priority over the port 0 setting. 192.168.1.4 will be asked to serve requests coming from *router1* to the local router. If the local router also has the same Telnet and FTP entries from the previous example, 192.168.1.3 will serve the Telnet request, 192.168.1.2 will serve the FTP request, and 192.168.1.4 will serve any other request, including HTTP, SMTP, etc.

**Example 3:**

```
remote addServer 192.168.1.10 tcp 9000 9000 telnet route-in
remote addServer 192.168.1.11 tcp 9001 9001 telnet route-in
```

In this example, an incoming request on TCP port 9000 will be sent to 192.168.1.10 with the port changed from 9000 to the Telnet (port 23).

An incoming request on TCP port 9001 will be sent to 192.168.1.11 with the port changed from 9001 to the telnet port.

**"Failed to add server" error message**

The error message *Failed to add server* is printed if a server entry could not be created. This can occur for two reasons:

**Port overlap:** One or more of the ports would be visible to the remote end overlap. For example, you enter:

```
remote addserver 192.168.1.10 tcp 9000 9000 telnet router1
```

Let us assume this command is accepted. Then you enter:

```
remote addserver 192.168.1.11 tcp 9000 9000 telnet router1
```

Let us assume this command gets an error. If the remote end sends a server request to port 9000, it cannot know to which server, 192.168.1.10 or 192.168.1.11, to send the request, if both entries exist.

**Not enough memory was available to create an entry.** This condition should not ordinarily occur because the amount of memory needed for a server entry is less than 30 bytes. Should this problem occur, it may cause many related problems or failures.

- ## System Commands

The following two commands are used to globally enable/disable a local IP address (on your LAN) as the server for that particular protocol.

**system addServer** <*ipaddr*> discard|me <*protocolid*> tcp|udp <*first port*> ftp|telnet|smtp|snmp|http [<*last port*>[<*first private port*>]]

**system delServer** <*ipaddr*> discard|me <*protocolid*> tcp|udp <*first port*> ftp|telnet|smtp|snmp|http [<*last port*>[<*first private port*>]]

where

*first port:* this is the first or only port as seen by the remote end.

*last port:* if specified, it is used with <*first port*> to specify a range of ports as seen by the remote end for the server on your LAN.

*first private port:* if specified, this is a port remapping of the incoming request from the remote end.
*first port* maps to *first private port*.
*first port* + 1 maps to *first private port* + 1

*last port* maps to *first private port + last port - first port*

*first port* through *last port* are the ports as seen by the remote end.
*first private port* through *first private port + last port - first port* are the equivalent ports the server on your local LAN will receive the request.

Remember to type **save** to make the changes persistent across boots.

**Examples:**

```
system addserver 192.168.1.5 tcp smtp
system addserver 192.168.1.6 tcp 0
system addserver 192.168.1.6 udp 0
```

The router sends a server request for SMTP to 192.168.1.5 when such a request comes from any remote router running NAT. The router sends any other server request (tcp or udp) to 192.168.1.6.

## • Server Request Hierarchy

When handling a request from a remote router (to which the local router has NAT enabled), the local router selects a server based on the following priority algorithm:

1. **remote addserver** — The local router selects a server for the remote router that handles that particular protocol/port.

2. **system addserver** — The local router selects a global server that handles that particular protocol/port.

3. **remote addserver** with *port* 0 — The local router selects a server for the remote router that handles that particular protocol (such as tcp/udp) and *any* port.

4. **system addserver** with *port* 0 — The local router selects a global server that handles that particular protocol and *any* port.

5. If an IP address is used for true NAT host remapping as well as for IP address/port translation, the IP address of the local remapped host as the server is selected.

6. Router's **IP address** — The local router selects itself (the local router) as the server.

## Classic NAT

With classic NAT, one PC IP address is translated to one NAT IP address. This NAT technique is primarily used to make certain hosts on a private LAN globally visible and give them the ability to remap these IP addresses as well.

## Client Configuration

Classic NAT requires that you first enable NAT Masquerading (as described in the previous section); thus, for the Classic and Masquerading forms of NAT, the clients are configured in the same way. Refer to the *Client Configuration, page 91* section.

### Host Remapping

- #### Remote Commands

  Use the following two commands to enable or disable host remapping on a per-remote basis:

  **remote addHostMapping <***first private addr***> <***second private addr***> <***first public addr***> <***remoteName***>**

  **remote delHostMapping <***first private addr***> <***second private addr***> <***first public addr***> <***remoteName***>**

  Use the command **remote addHostMapping** whenever a host on the local LAN is known by different IP addresses to different remotes.

- #### System Commands

  Use these commands to enable or disable host remapping systemwide:

  **system addHostMapping** *<first private addr>* **<***second private addr***> <***first public addr***>**
  **system delHostMapping <***first private addr***> <***second private addr***> <***first public addr***>**

  Use the command **system addHostMapping** whenever a host on the local LAN is known by the same IP address on all remotes.

- #### IP Address Range

  The range of local LAN IP addresses to be remapped is defined by *<first private addr>* to *<second private addr>* inclusive. These addresses are mapped one-to-one to the public addresses.

  The range of public IP addresses is defined by *<first public addr>* only. The rest of the range is computed automatically (from *<first public addr>* to *<first public addr>* + number of addresses remapped - 1) inclusive.

- ## Multiple-Host Remapping Entries

  Users may enter as many host remapping entries as they wish.

  **Example:**

  ```
  remote addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11
  remoteName
  remote addHostMapping 192.168.207.93 192.168.207.99 10.0.20.4
  remoteName
  remote addHostMapping 192.168.209.71 192.168.209.80 10.12.14.16
  remoteName
  ```

  The above entries create three mappings:

  192.168.207.40 through 192.168.207.49 are mapped to 10.0.20.11 through 10.0.20.20
  192.168.207.93 through 192.168.207.99 are mapped to 10.0.20.4 through 10.0.20.10
  192.168.209.71 through 192.168.209.80 are mapped to 10.12.14.16 through 10.12.14.25

- ## Range Overlap Rules

  With **remote addHostMapping,** private IP address ranges cannot overlap for a remote router.
  With **remote addHostMapping**, public IP address ranges cannot overlap for a remote router.

  With **system addHostMapping**, private IP address ranges cannot overlap for a system.
  With **system addHostMapping**, public IP address ranges cannot overlap for a system.

  If a private IP address range for a remote router and a private IP address range for the system overlap, the private IP address range for the remote has precedence.

  If a public IP address range for a remote and the public IP address range for the system overlap, the public IP address range for the remote has precedence.

  Private IP addresses and public IP addresses can be the same.

  For example, to enable IP/port translation to a remote router and make the IP addresses 10.1.1.7 through 10.1.1.10 globally visible, it is permissible to use either one of the following commands:

  ```
  remote addHostMapping 10.1.1.7 10.1.1.10 10.1.1.7 remoteName
  system addHostMapping 10.1.1.7 10.1.1.10 10.1.1.7
  ```

  If the remapped host's IP address (classic NAT, one-to-one IP address translation) and the masquerading IP address (many-to-one IP address translation) are the same, then NAT masquerading has precedence over classic NAT.

# Controlling Remote Management

With the following security control features, the user can control remote management of the router via Telnet, HTTP, Syslog, and/or SNMP. Disabling SNMP stops the Configuration Manager from accessing the router, which in some environments is desirable.

Router system event messages can be automatically sent to a Unix Syslog server. The **system syslogport** and **system addsyslogfilter** commands control the access and port numbers.

## Disabling Remote Management

To completely disable remote management, enter the following commands from the command line:

**system telnetport disabled**
**system snmpport disabled**
**system httpport disabled**
**system syslogport disabled**
**save**
**reboot**

## Re-enabling Remote Management

To reestablish the disabled remote management services, restore the default values with the commands:

**system telnetport default**
**system snmpport default**
**system httpport default**
**system syslogport default**

## Validating Clients

The following commands are used to validate clients for Telnet, SNMP, HTTP, or Syslog. They define a range of IP addresses that are allowed to access the router via that interface. Only the IP addresses in the range specified for the interface can access the router via that interface. This validation feature is **off** by default.

Multiple ranges can be specified for Telnet and SNMP clients. If no range is defined, then access to the router is through the LAN or WAN.

**Note:** These commands do *not* require a reboot and are effective immediately.

**system addtelnetfilter <***first ip addr***>** [**<***last ip addr*>] | LAN
**system addsnmpfilter <***first ip addr*> [<*last ip addr*>] | LAN
**system addhttpfilter <***first ip addr*> [<*last ip addr*>] | LAN
**system addsyslogfilter <***first ip addr*> [<*last ip addr*>] | LAN

| | |
|---|---|
| *first ip addr* | First IP address of the client range |
| *last ip addr* | Last IP address of the client range. May be omitted if the range contains only one IP address. |
| LAN | Local Ethernet LAN |

**Example:**

```
system addsnmpfilter 192.168.1.5 192.168.1.12
```

To delete client ranges previously defined, use these commands:

**system deltelnetfilter** <*first ip addr*> [<*last ip addr*>] | LAN
**system delsnmpfilter** <*first ip addr*> [<*last ip addr*>] | LAN
**system delhttpfilter** <*first ip addr*> [<*last ip addr*>] | LAN
**system delsyslogfilter** <*first ip addr*> [<*last ip addr*>] | LAN

To list the range of allowed clients, use the command:

**system list**

## Restricting Remote Access

To allow remote management while making it more difficult for non-authorized persons to access the router, you may redefine the ports to a less well-known value. When Network Address Translation (NAT) is used, this port redefinition feature also allows you to continue using the standard ports with another device on the LAN (provided the appropriate NAT server ports commands are issued), while simultaneously managing the router (with non-standard ports).

For example, the following commands redefine the Telnet, SNMP, HTTP, and Syslog ports:

```
system telnetport 4321
system snmpport 3214
system httpport 5678
system syslogport 6789
```

## Changing the SNMP Community Name

Changing the SNMP community name from its default value of “public” to another string may further enhance SNMP security. This string then acts like a password, but this password is sent in the clear over the WAN/LAN, in accordance with the SNMP specification.

Use the following commands to change the SNMP community name.

**system community** <*new community name*>
**save**
**reboot**

## Disabling WAN Management

You can allow management of the router on the local LAN, but not over the WAN. If the router has been configured to use NAT, you can define two servers that *do not* exist on the LAN side to handle WAN SNMP and Telnet requests, and thus WAN management of the router cannot occur.

The following example shows how this is done. It assumes there is no computer at 192.168.254.128.

```
system addServer 192.168.254.128 udp snmp
system addServer 192.168.254.128 tcp telnet
system addServer 192.168.254.128 tcp http
save
reboot
```

# Software Option Keys

This router has several optional software features that can be purchased as software option keys when ordering the router. These optional features are:

- IP routing
- DES or 3DES encryption (see *Encryption, page 100*)
- IP filters (see *IP Filtering, page 103*)
- L2TP tunneling (see *L2TP Tunneling — Virtual Dial-Up, page 106*)
- IPSec (see IPSec (Internet Protocol Security), page 119)

These options are usually ordered with the router. The options are controlled by the presence of a key file in flash memory or a bit set in the CMOS. Both values are checked; if either is set, the option is enabled.

## Listing the Installed Software Options

To determine which software options are installed in your router, use the **vers** command. The **vers** command lists all options:

- Options that are disabled are shown with a ~ prefix.
- Options that are enabled by a key file have a + prefix.
- Options that are enabled by a set bit have no prefix.

For example, the following **vers** command output shows that the L2TP and encryption options are disabled, but all the other options are enabled.

```
Options: FRAME RELAY, SDSL, RFC1490, +IP ROUTING, IP FILTERING, WEB, ~L2TP,
 ~ENCRYPT, BRIDGE, IPX, CMMGMT
```

## Adding a New Software Option Key

A software option key is a 44-character string, unique to a particular router, that enables a single feature. After receiving a software options key, you can enter it using either the web GUI or the Command Line Interface.

When using the web GUI, you select the Upgrade Features button and enter the key. When using the Command Line Interface, you enter the key using the following command:

**key add** <keystring>

**Note:** The new feature is not activated until the router is rebooted.

# Encryption

**Note:** Encryption is a software option. The following section applies only for routers with this option.

For routers shipped with the following encryption options, two variants of encrypted data links over PPP have been implemented:

- PPP DES (Data Encryption Standard) (RFC1969)
- Diffie-Hellman

Encryption requires PPP.

**Caution:** PPP DES and Diffie-Hellman encryption options may not be exported outside the United States or Canada.

## PPP DES (RFC 1969) Encryption

PPP DES (Data Encryption Standard) implementation uses a 56-bit key with fixed **transmit** and **receive** keys that are specified in each router. With RFC 1969, users must manage the keys. This implementation has been tested for interoperability with other PPP DES vendors such as IBM and Network Express (part of Cabletron).

### Configuration Notes

Simply add the encryption commands to your standard configuration. For PPP DES, the encryption commands are:

**remote setEncryption dese rx <***key***> <***remoteName***>**
**remote setEncryption dese tx <***key***> <***remoteName***>**

Observe the following guidelines:

- PPP DES can only be configured using the Command Line Interface (CLI).
- The choice of keys should be carefully considered: they must have eight hexadecimal digits, and values that are considered cryptographically weak should be avoided. Consult a security expert for advice.
- Use the console port or a Telnet port (use the **system log** command) to view error messages and progress. If you see “Unknown protocol” errors, the router **receive** key and **sender Tx** key don't match.
- Different keys may be used with different remote destinations.
- For maximum security, as shown in the following configuration examples, Telnet and SNMP access should be disabled, and PPP CHAP authentication should be used by both ends.

### Sample Configuration

Refer to the section *Sample Configurations, page 59,* of this guide. The routers SOHO (the target router) and HQ (the remote router) are configured in the same manner as shown in that section, but the following encryption commands are added. Don’t forget to save the configuration and reboot the router (**save** and **reboot** commands).

Remember that the **transmit** key (**tx**) of SOHO is the **receive** key (**rx**) of HQ. Inversely, the **receive** key of SOHO is the **transmit** key of HQ.

Use this sample configuration with the additional encryption commands as a guideline to configure your own routers.

- **Enable Encryption on the Router HQ**

```
Example:
remote setEncryption dese rx 1111111111111111 SOHO
remote setEncryption dese tx 2222222222222222 SOHO
save
reboot
```

- **Enable encryption for the router SOHO**

```
Example:
remote setEncryption dese tx 1111111111111111 HQ
remote setEncryption dese rx 2222222222222222 HQ
save
reboot
```

## Diffie-Hellman Encryption

With Diffie-Hellman encryption, each router has an encryption file that is associated with a public key providing 768-bit security. The predefined keys can be replaced by the user. The key files have a suffix of “num” by convention (e.g., dh96.num).

### Configuration Notes

Simply add the encryption command to your standard configuration. For Diffie-Hellman, the encryption command is:

**remote setEncryption** DESE_1_KEY|DESE_2_KEY [<*fileName*>]| <*remoteName*>

Observe the following guidelines:

- DESE_1_KEY specifies that the same key is used in both directions, whereas DESE_2_KEY specifies that the keys are different. Having the same keys in both directions can significantly reduce time needed to compute the DES keys from the Diffie-Hellman exchange.
- The routers’ **receive** key and **sender** Tx key must not match.
- Different keys and key files may be used with different remote destinations.
- For maximum security, as shown in these examples, Telnet and SNMP access should be disabled, and PPP CHAP should be used. Use the console port to view error messages and progress.

### Sample Configuration

The sample configuration is the same as the one provided in the preceding PPP DES encryption example, but the Diffie-Hellman encryption command is used instead of the PPP DES encryption commands.

```
remote setEncryption DESE_1_KEY dh96.num SOHO
save
reboot
```

## File Format for the Diffie-Hellman Number File

The file consists of 192 bytes, in binary format. There are two 96-byte numbers stored, with the most significant byte in the first position. For example, the number 0x12345678 would appear as 000000...0012345678.

The first 96 bytes form the modulus. In the equation *x' = g^x mod n*, *n* is the modulus. According to Diffie and Hellman, the modulus should be prime, and *(n-1)/2* should also be prime.

The second 96 bytes form the generator, or g in the above equation. The generator should be a primitive root mod *n*.

The remaining pieces of the encryption key (x and y) are randomly generated at connection time and will change every time the device connects.

Contact an encryption expert to obtain cryptographically sound generator and modulus pairs if you wish to change the default values.

- ### Default Modulus

```
00000000: c9 b4 ed 33 ba 7f 00 9e - ce e0 83 5d a5 4c 19 25
00000010: e0 2d 99 44 e8 8d cd 16 - 02 0e 6c 26 6d 15 7c 95
00000020: 82 9a 8c 2b 19 d0 56 da - 9b 5b a9 cd cf fb 45 2b
00000030: c9 6a 3c 26 e5 b8 1a 25 - 07 b8 07 22 ed 15 8a 56
00000040: 8b f4 30 f2 28 fc 6b f1 - bf a4 3e 87 f0 be d6 1c
00000050: 33 92 b9 5e d1 b7 20 8c - 92 02 cb e5 26 45 02 1d
```

- ### Default Generator

```
00000000: 90 f0 09 78 cc 23 79 a8 - 6c 23 a8 65 e0 dc 0f 6d
00000010: fb a7 26 e8 63 0a 21 67 - 5a f8 0f 59 84 09 5c da
00000020: ef af af fc d2 5f 83 e2 - a7 27 05 34 17 94 1a 4f
00000030: b2 87 76 97 e7 48 43 db - 62 29 70 9e 7f eb 2c 6e
00000040: 5d 25 1d a1 65 f0 b4 e6 - 47 4d 25 23 0b 20 b9 93
00000050: 27 f0 56 12 5a 97 f6 c5 - 31 b6 19 fc 67 22 93 f5
```

# IP Filtering

IP Filtering is a type of firewall used to control network traffic. The process involves filtering packets received from one interface and deciding whether to route them to another interface or discard them.

When it is filtering packets, the router examines information such as the source and destination address contained in the IP packet, the type of connection, etc., and then screens (filters) the packets based on this information; packets are either allowed to be forwarded from one interface to another interface or simply discarded.

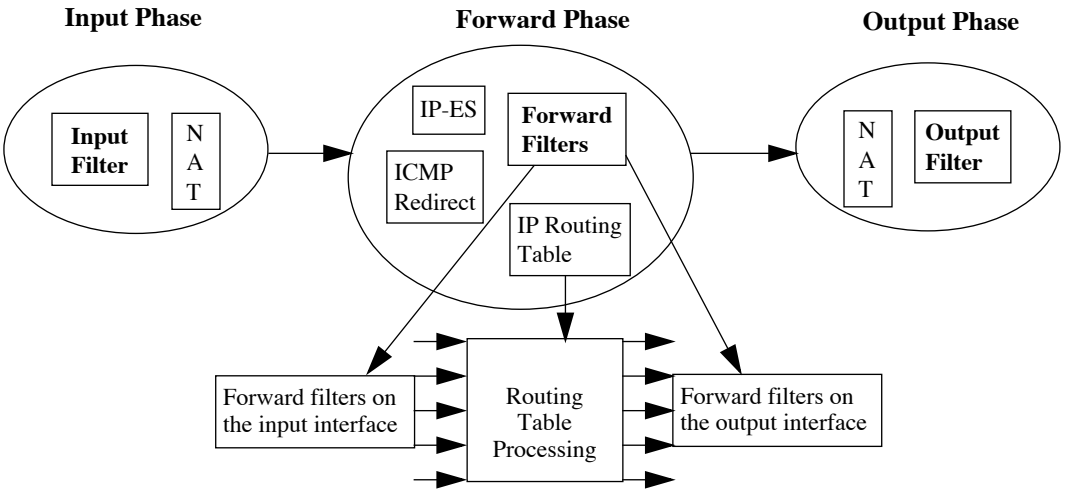
IP filtering requires that IP routing be enabled. This type of filtering offers great flexibility and control of IP filters, but configuration of this feature requires using a series of commands that may appear complex to a casual user.

See the installation CD for sample IP Filter command files (**filter.txt**). This file can be edited for your installation and copied to the router usingTFTP or the Windows Quick Start application. (See Batch File Command Execution, page 279.)

## Filters and Interfaces

Filters are commands used to screen IP packets: packets are simply matched against a series of filters. If a packet matches a filter, the filter determines whether the packet is accepted or dropped. If no filter matches the incoming packet, the packet is accepted by default.

Filters operate at the interface level. Each interface has a series of IP filters associated with it; the filters can be of three types: Input filters, Output filters, and Forward filters. The following illustrates the filter process.




### Input Phase

When an IP packet comes in through an interface (such as the Input interface), the router tries to recognize the packet. The router then examines the Input filters for this interface and, based on the first Input filter that matches the IP packet, it decides how to handle the packet (forward or discard it).

If NAT translation is enabled for the Input interface, NAT translation is performed.

### Forward Phase

At this stage, the router uses its routing table to determine to which interface or link the packet is sent . It then applies the Forward filters based on the Input interface information. Next the router applies the Forward filters based on the Output interface information.

### Output Phase

If NAT translation is enabled for the Output interface, then NAT translation is performed. The router examines the Output filters for this interface andhandle the IP packet based on the first Output filter that matches the packet.

## Configuring Filters with Network Address Translation Enabled

### General NAT Information

Network Address Translation is an IP address conversion feature that translates a PC's local (internal) address into a global (outside/Internet) IP address. NAT is needed when a PC (or several PCs) on a Local Area Network wants to connect to the Internet or get to a remote network that uses global, registered addresses: NAT swaps the local IP address with a global IP address: the IP address and port information that the PC uses are remapped (changed) to the IP address that was assigned to the router and a new port number is assigned.

**Note:** The preceding section, *Filters and Interfaces*, describes how NAT "behaves" for each filtering phase.

## Filter Actions

Each packet is compared to filters at each implementation point (Input, Forward, and Output). If *no* filter at a particular point matches the incoming IP packet, the packet is accepted. If a filter does match the packet, the filter initiates one of the following possible actions:

### Accept

When the packet is accepted at a filter interface (Input, Forward, or Output), the router lets it proceed for further processing.

### Drop

With Drop, the packet is discarded.

### Reject

With Reject, an ICMP REJECT (Internet Control Management Protocol) is sent to reject the packet.

### Pass to IPSec

Two actions—**inipsec** and **outipsec**—pass the packet to IPSec for further processing. The **inipsec** action is for packets coming from the other IPSec gateway; it passes the packet to IPSec for decrypting. The **outipsec**

action is for packets coming from the local protected network; it passes the packet to IPSec so it can be encrypted and sent to the other IPSec gateway.

Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE, rather than your own filters, to manage your IP Security (see IPSec (Internet Protocol Security), page 119).

## IP Filter Commands

To define and manage IP filters on an Ethernet interface, use the command **eth ip filter**. To define and manage IP filters on the remote interface, use the command **remote ipfilter**. For more information on these commands, see ETH IP FILTER, page 172 and REMOTE IPFILTER, page 191.

## Special Notes

IP filters of Input type are checked *before* the IP packet is redirected by ICMP. This could adversely affect local LANs that use ICMP redirect to dynamically learn IP routes. IP filters of Input type are checked *before* the IP packet is sent to the router itself as a host.

**Example:**

The following commands stop *any* attempt by a host coming from the remote Internet from sending an IP packet to the Telnet port. Hence, the router does not see the packet, and the packet is not forwarded.

```
remote ipfilter insert input drop -p tcp -dp 23 internet
save
```

These commands stop *any* attempt by a host coming from the remote internet from sending an IP packet to the Telnet port “through” the router to a different interface. The router itself could still receive the IP packet, hence the remote host could Telnet to the router itself.

```
remote ipfilter insert forward drop -p tcp -dp 23 internet
save
```

# L2TP Tunneling — Virtual Dial-Up

This section has four parts:

- The *Introduction* provides a general overview of L2TP tunneling.
- The *L2TP Concepts* section explains LNS, L2TP client, LAC, dial user, tunnels, and sessions.
- *Configuration* describes preliminary configuration steps and verification steps and lists commands associated with the configuration of L2TP and PPP sessions.
- The *Sample Configurations* section provides two examples with step-by-step instructions: a simple L2TP client configuration example and a complete LNS and L2TP client configuration example.

  The installation CD also contains sample configuration files. These files can be edited for your installation and copied to the router using TFTP or the Windows Quick Start application. For more information on TFTP use, see Batch File Command Execution, page 279.

## Introduction

L2TP  (Layer 2 Tunneling Protocol) is used to forward a PPP link from a remote site to a corporate site across the Internet, thus creating virtual paths called tunnels. Because tunneling involves encapsulating data, packets can be transported across networks using different protocols. The advantages for tunneling the PPP protocol are listed below:

- Different network protocols such as NetBEUI, IPX, and Appletalk can be transported through the Internet using a tunnel. The protocol packets are encapsulated and routed across the network through the Internet.
- Tunnels provide a way to reduce costs and complexity associated with remote dial-up networking by using a local ISP: users connect to the remote site by dialing into their local ISP and letting the Internet handle the long-distance connections, thus avoiding long-distance phone charges.
- Tunneling PPP allows compression of data through the entire tunnel, which translates into greater throughput.
- By allowing encryption over the PPP link, L2TP contributes to more secure networks over the Internet.
- Remote users can access the company network, even if there is a company firewall (provided, of course, that tunnels can come through the firewall).

**Note:** This feature can interoperate with any vendor that supports L2TP - Draft II.

## L2TP Concepts

This section defines the major L2TP concepts such as LNS, L2TP client, LAC, and Dial user. These concepts are illustrated with L2TP client examples. Also described are tunnels and sessions' creations and destructions.

## LNS, L2TP Client, LAC, and Dial User

An L2TP tunnel is created between an L2TP client and LNS. The L2TP client and LNS control the tunnel using the L2TP protocol.

Since routers are more often configured as L2TP clients or LNS than as LACs, this section, therefore, emphasizes L2TP client- and LNS-related information.

- ### LNS (L2TP Network Server)

  The LNS is the point where the call is actually managed and terminated (e.g., within a corporate network).

- ### L2TP Client

  With an L2TP client, the dial user and LAC are combined in the same hardware device. In this case, the PPP session is between the LAC and the LNS.

  As shown in the following illustration (Figure 1), an L2TP client is used to tunnel a PPP session between a small office (our router) and a corporate office through the Internet.

- ### LAC (L2TP Access Concentrator)

  The LAC can be envisioned as the physical hardware (e.g., a router) used for placing and receiving phone calls.

- ### Dial User

  A dial user is the remote system or router that is either placing the call to the LAC or receiving the call from the LAC.

  The dial user does not actually dial in to the LNS or receive a call from the LNS, since this is a virtual connection.

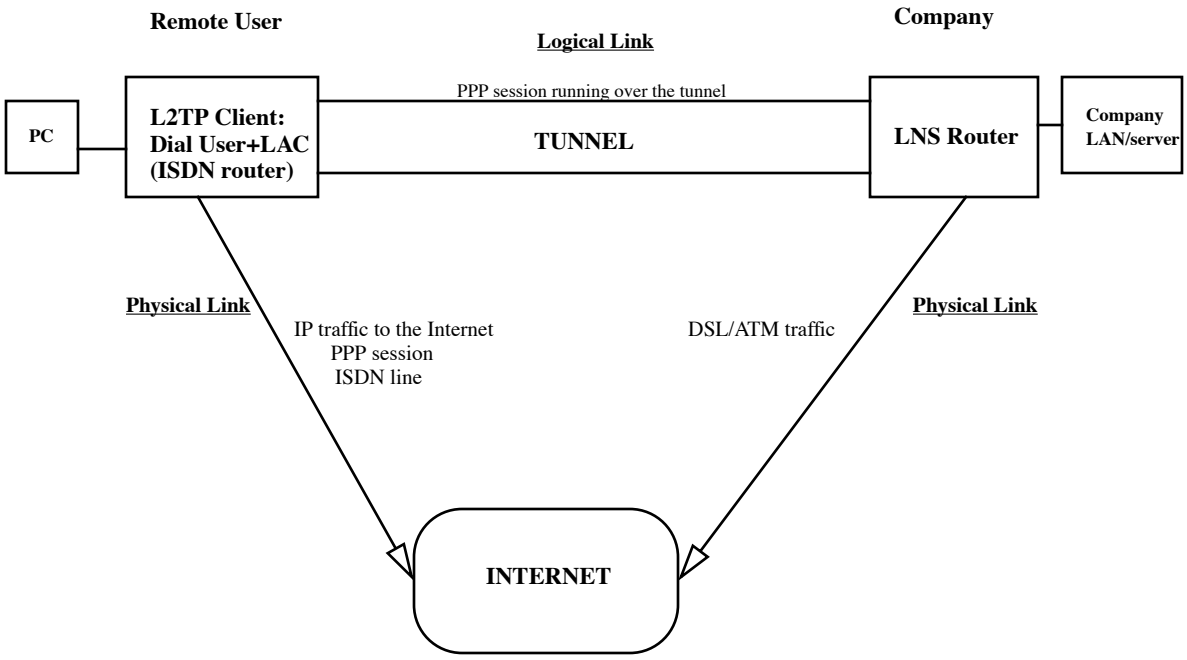
  The dial user is one end of a PPP session. The LNS is the other end of the PPP session.

## L2TP Client Example

The tunnel uses UDP/IP traffic as the transport medium over IP. This implementation of L2TP as illustrated below shows a tunnel from a remote user's perspective.

**Note:** There is one PPP session over ISDN and another PPP session over the tunnel.

**Figure 1**




## LNS and L2TP Client Relationship

The LNS acts as the supervising system. The L2TP client acts both as the dial user and the LAC.

One end of the tunnel terminates at the L2TP client. The other end of the tunnel terminates at the LNS.

One end of the PPP session going through the tunnel terminates at the L2TP client acting as the dial user; the other end terminates at the LNS.

## Tunnels

Tunnels are virtual paths that exist between an L2TP client and LNS.

An LNS can communicate simultaneously with more than one L2TP client.

An L2TP client can communicate simultaneously with more than one LNS.

Some L2TP implementations including the one discussed in this section allow the *same* router to act as *both* an L2TP client and LNS simultaneously, if so configured.

**Caution:** Verify that the IP address of the other end of the tunnel is correctly routed through the right, local interface/remote and will not appear to be routed through the tunnel. An attempt to route the tunnel endpoint within itself will fail.

### Sessions

Sessions can be thought of as switched virtual circuit “calls” carried within a tunnel and can only exist within tunnels. One session carries one “call”. This “call” is one PPP session. Multiple sessions can exist within a tunnel. The following briefly discusses how sessions are created and destroyed.

- #### Session creation

  Traffic destined to a remote entry (located at the end of the tunnel) will initiate a tunnel session. When the L2TP client wishes to establish a session to an LNS, the L2TP client assumes the role of a LAC and sends control packets containing incoming call information to the LNS over the tunnel.

- #### Session destruction

  A tunnel session will automatically time out after the data session stops. When instructed to destroy a session, the L2TP client closes any PPP session associated with that session. The L2TP client may also send control messages to the LNS indicating that the L2TP client wishes to end the PPP session.

  When the LNS wants to hang up the call, it sends control messages destroying the session.

## Configuration

### Preliminary Steps to Configure a Tunnel

The following logical steps should be considered before configuring a tunnel:

1. Decide if the router will act as an L2TP Client or LNS.
2. Decide if one side or both sides of the connection can initiate a tunnel.
3. Create the L2TP Tunnel Entry with these characteristics:
   - An L2TP client host name
   - An LNS host name
   - A Tunnel CHAP secret (both sides of the connection must use the same secret)
   - The IP address of the other party must be provided to the initiating side of the tunnel
   - Type of flow control (pacing, sequence numbers or not)
4. Create a remote entry for the PPP session. Associate the remote entry with the Tunnel.

### Verification Steps

1. Verify that the IP address of the other end of the tunnel is correctly routed through the right, local interface/remote and will not appear to be routed through the tunnel. An attempt to route the tunnel endpoint within itself will fail.
2. Try to establish IP connectivity (using the **ping** or **tracert** commands).

a.“Pinging” from the L2TP client or LNS to the opposite tunnel endpoint will succeed (this tests the tunnel path).

b.“Pinging” from a tunnel endpoint IP address to an IP address within the tunnel will probably fail due to the existence of the IP firewall.

## Configuration Commands

There are two categories of L2TP commands which are respectively associated with:

- Tunnels and the L2TP protocol
- The PPP session

### • Commands associated with tunnels and the L2TP Protocol

These commands are used to configure L2TP tunnels. For additional information on the syntax of the commands listed below, please refer to the L2TP commands section in the Command Line Interface Reference chapter.

#### L2TP tunnel entry:

**l2tp add** <*TunnelName*>

#### The remote tunnel host name:

**l2tp set remoteName** <*name*> <*TunnelName*>

#### The local tunnel host name:

**l2tp set ourTunnelName** <*name*> <*TunnelName*>

#### CHAP Secret:

**l2tp set CHAPSecret** <*secret*> <*TunnelName*>

#### Tunnel Authentication:

**l2tp set authen on|off** <*TunnelName*>

#### Type of L2TP support for tunnel:

A tunnel entry can be configured to act as a LAC, an LNS, both a LAC and LNS, or disabled.

**l2tp set type all|lns|l2tpclient|disabled** <*TunnelName*>

#### Remote tunnel IP address:

**l2tp set address** <*ipaddr*> <*TunnelName*>

**Note:** Verify that the IP address of the other end of the tunnel is correctly routed. It should not be routed through the tunnel itself, but over a physical link.

#### Our PPP system name and secret/password:

The following commands specify the router’s name and password/secret for authentication purposes on a per-tunnel basis.

**l2tp set ourSysName** <*name*> <*TunnelName*>
**l2tp set ourPassword** <*password*> <*TunnelName*>

**Miscellaneous commands:**

Commands used to delete a tunnel, close a tunnel, or set up advanced L2TP configuration features such as traffic performance fine-tuning are discussed in the *L2TP Commands* section of Chapter 5.

### PPP Session Configuration

Two commands are used to extend a PPP link from a remote site to a corporate site across the Internet and establish a tunnel. For additional information on the syntax of the commands listed below, refer to the *Remote Commands* section of Chapter 5.

**remote setLNS** <*TunnelName*> <*remoteName*>

**remote setl2tpclient** <*TunnelName*><*remoteName*>

## Sample Configurations

Two sample configurations are described in this section:

- A simple configuration. This example describes the information needed to configure one side of the tunnel (the client side).
- A complete configuration. This example describes the information needed to configure both sides of the tunnel (client and server sides).

### Simple L2TP Client Configuration Example

This example shows how a telecommuter working at home (client side) can configure his/her router SOHO to tunnel to the company's LAN (server side).

The information given in the Configuration Process section below provides a framework reference for this type of L2TP Client configuration.

#### • Assumptions

In this example, the following information is assumed:

- The server side (the company) has an LNS router connected to the Internet.
- The client side has an existing route to the Internet with the remote "Internet" (refer to the following Note, if you need sample configuration commands).
- IP routing is enabled (refer to the following Note, if you need sample configuration commands).

**Note:** Below is an example of configuration commands that can be used to enable IP routing and establish a route to the Internet.

```
remote add internet
remote disauthen internet
remote setoursysname name_isp_expects internet
remote setourpass secret_isp_expects internet
remote addiproute 0.0.0.0 0.0.0.0 1 internet
remote setphone isdn 1 5551000 internet
remote setphone isdn 2 5553000 internet
eth ip enable
eth ip address 192.168.254.254 255.255.255.0
```

## • Configuration Process

The following sets of questions, answers, and configuration commands specific to the L2TP tunnel and the PPP remote will assist you in configuring the client side router SOHO (also referred to as home router). Note that the server side is referred to as either company router or router at work.

### L2TP tunnel configuration

#### L2TP tunnel-specific questions

1. What is the host name of the router at home that the user is configuring?
2. What is the host name of the company router at work to which the user will tunnel?
3. What is the shared CHAP secret used for tunneling between the home router (client) and the company router (server)?
4. What is the IP address of the company router to which the user will tunnel?

**L2TP tunnel answers.** For our example, let's assume the answers to the above tunnel-specific questions are as follows:

1. Home_Router
2. Work_Router
3. Shared_Secret
4. 10.0.0.1

**L2TP tunnel configuration commands.** These commands would be used to set up the L2TP tunnel information for our example:

```
l2tp add Work_Router
l2tp set ourtunnel Home_Router Work_Router
l2tp set chapsecret Shared_Secret Work_Router
l2tp set address 10.0.0.1 Work_Router
```

### PPP remote configuration

**PPP remote-specific questions:**

1. What is the home router's name for PPP authentication?
2. What is the home router's secret for PPP authentication?
3. Does the home router need PPP authentication for the remote router (company router)?

   If yes:

   a. What is the remote router's name for PPP authentication?
   b. What is the remote router's secret for PPP authentication?

   If no:

   a. Use the command **remote disauthen** <*remoteName*> where <*remoteName*> is the name used to refer to the company's router.

4. Does the remote router dynamically assign an IP address for this PPP session?

   If *yes*:

   Use IP address translation (NAT)

   If *no* and the home router is to behave as a LAN at home:

   Which IP address and network mask does the home router use for its LAN at home? Use the **eth ip addr** command to set the LAN at home. Do not enable IP address translation (NAT) for the remote (company) router.

   If *no* and the home router is to behave as a host at home:

   Which IP address does it use at home? Assuming an IP address of www.xxx.yyy.zzz, use the command:

   **remote setsrcipaddr www.xxx.yyy.zzz 255.255.255.255** <*remoteName*>

   **remote setiptranslate on** <*remoteName*>

5. Which IP and network addresses does the home router access at work through this PPP session?

**PPP remote answers.** For our example, let us assume the answers to the above PPP remote-specific questions are as follows:

1. ppp_soho
2. ppp_soho_secret
3. We assume that this router will authenticate the router at work with the following information:

   a) the company router's name is: ppp_work
   b) the company router's PPP secret is: ppp_work_secret

4. We assume that the company's router will dynamically assign an IP address to the home router.
5. 172.16.0.0/255.240.0.0

**PPP remote configuration commands.** For our example, these commands would be used to set up the PPP remote information for tunneling to work:

```
remote add ppp_work
remote setlns Work_Router ppp_work
remote setpasswd ppp_work_secret ppp_work
remote setiptranslate on ppp_work
remote addiproute 172.16.0.0 255.240.0.0 1 ppp_work

l2tp set oursysname ppp_soho Work_Router
l2tp set ourpassword ppp_soho_secret Work_Router
```

## Complete LNS and L2TP Client Configuration Example

The following information and illustration (Figure 2) provide a configuration example of an LNS and L2TP Client.

- ### Assumptions

#### IP Addresses

The LNS server's LAN IP address is 192.168.100.1 (**LNSserver**) with a mask of 255.255.255.0.

The LNS has a WAN IP address of 192.168.110.1, which is used as the tunnel endpoint.

The LNS connects to the remote **internet**.

The L2TP Client's LAN IP address is 192.168.101.1 (**soho**) with a mask of 255.255.255.0. Additionally, 192.168.101.1 is also the tunnel endpoint within the L2TP client. The router **soho** connects to the remote **isp**.

#### Secret/password

A shared tunnel secret of "tunnelsecret" will be used.

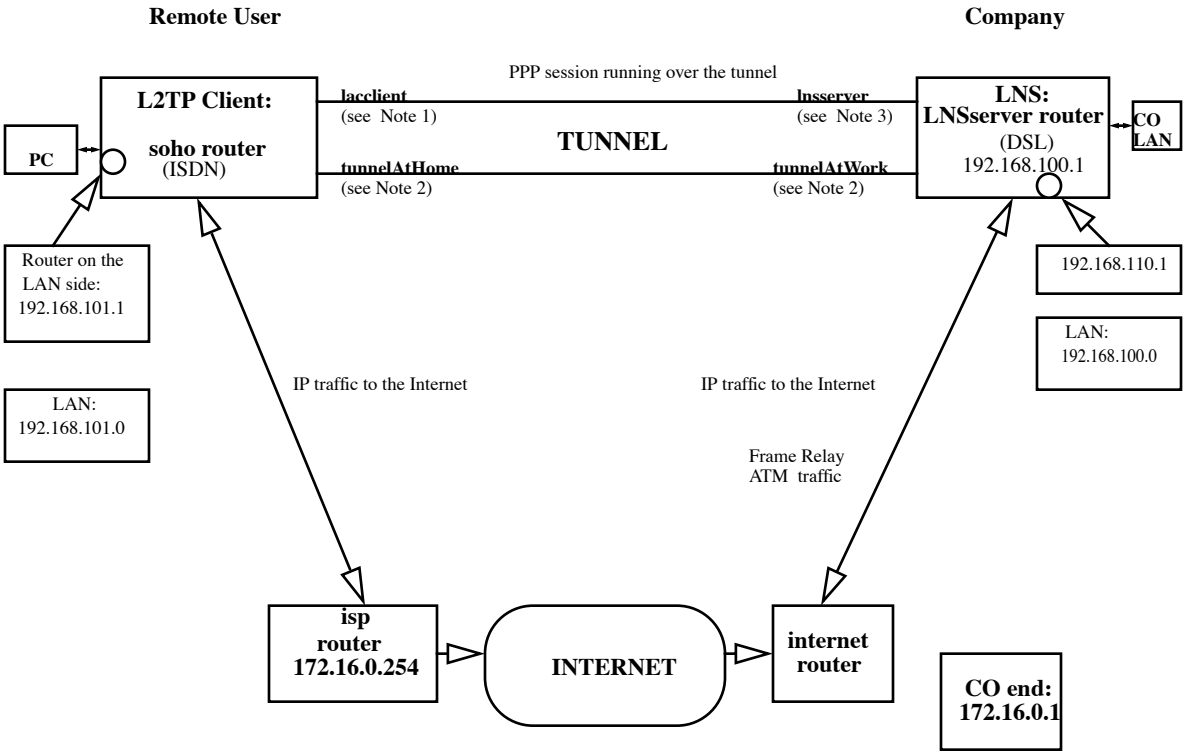
#### PPP Authentication

The LNS will authenticate the client using PPP. The client will not try to authenticate the LNS using PPP. For PPP authentication, the L2TP client will be known as "lacclient" with a password of "clientpassword".

#### Tunnel

Only the L2TP client (**soho**) will initiate the tunnel and make the connection. The tunnel is routed through the remote **internet** which is the default route. The LNS server never calls the L2TP client (**soho**).

**Figure 2**




**Note 1:** The CHAP secret is “clientPassword”.

**Note 2:** The CHAP secret is “tunnelSecret”.

**Note 3:** No CHAP secret is needed; the client does not authenticate the LNS server.

## Configuration Process

The following sample scripts list the commands used to configure the routers **soho** (L2TP client), **LNSserver** (LNS), **internet**, and **isp**.

- ### Configuration commands for soho (L2TP client)

  **Note:** soho is an ISDN router.

  **Define soho:**

```
system name soho
system passwd sohopasswd
system msg configured_12/15/98
system securitytimer 60
```

  **Enable IP routing for soho:**

```
eth ip enable
eth ip addr 192.168.101.1 255.255.255.0
```

**Set up ISDN parameters:**

```
isdn set switch ni1
isdn set dn 5551000 5553000
isdn set spids 0555100001 0555300001
```

**Define DHCP settings for DNS servers, domain, wins server:**

```
dhcp set value DOMAINNAMESERVER 192.168.100.68
dhcp set value DOMAINNAME flowpoint.com
dhcp set value WINSSERVER 192.168.100.73
```

**Define a remote for the tunnel:**

```
remote add lnsserver
remote disauthen lnsserver
remote setoursysname lacclient lnsserver
remote setourpasswd clientpassword lnsserver
remote setLNS tunnelAtWork lnsserver
remote addiproute 192.168.100.0 255.255.255.0 1 lnsserver
```

**Define a remote isp:**

```
remote add isp
remote setphone isdn 1 5552000 isp
remote setphone isdn 2 5554000 isp
remote disauthen internet remote addiproute 0.0.0.0 0.0.0.0 1 isp
```

**Define the tunnel:**

```
l2tp add tunnelAtWork
l2tp set chapsecret tunnelsecret tunnelAtWork
l2tp set ourtunnelname tunnelAtHome tunnelAtWork
l2tp set address 192.168.110.1 tunnelAtWork
save
reboot
```

- ## Configuration commands for internet

**Note:** internet is a DSL router. The router internet establishes a link to the LNS.

**Define internet:**

```
system name internet
system passwd internet
system msg configured_12/15/98
system securitytimer 60
```

**Enable IP routing and add routes:**

```
eth ip enable
eth ip addr 172.16.0.1 255.255.255.0
eth ip opt rxdef off
eth ip addroute 192.168.101.1 255.255.255.0 172.16.0.254 1
```

**Create a DHCP pool of addresses:**

```
dhcp add 172.16.0.0 255.255.255.0
dhcp del 192.168.254.0
dhcp set addr 172.16.0.2 172.16.0.20
```

**Set up DSL parameters:**

```
sd term co sd speed 1152
```

**Define a remote LNSserver**

```
remote add lnsserver
remote setauthen chap lnsserver
remote setpasswd serverpassword lnsserver
remote addiproute  192.168.110.1 255.255.255.255 1 lnsserver
remote setprotocol ppp lnsserver
remote setpvc 0*38 lnsserver
save
reboot
```

- ## Configuration commands for isp

**Note:** isp is an ISDN router. The router soho calls the router isp.

**Define isp:**

```
system name isp
system passwd isppasswd
system msg configured_12/15/98
system securitytimer 60
```

**Enable IP routing:**

```
eth ip enable
eth ip addr 172.16.0.254 255.255.255.0
```

**Add a route to the other end of internet:**

```
eth ip defgate 172.16.0.1
eth ip opt txdef off
```

**Disable DHCP:**

```
dhcp disable all
```

**Set up ISDN parameters:**

```
isdn set switch ni1
isdn set dn 5552000 5554000
isdn set spids 0555200001 0555400001
```

**Define a remote (soho):**

```
remote add soho
remote setauthen chap soho
remote setpassw sohopasswd soho
remote setphone isdn 1 5551000 soho
remote setphone isdn 2 5553000 soho
remote addiproute  192.168.101.0 255.255.255.0 1 soho
save
reboot
```

- ## Configuration commands for LNSserver

  **Note:** LNSserver is a DSL router.

  **Define LNSserver:**

```
system name lnsserver
system passwd serverpassword
system msg Script_for_LNS_called_HQ
system securitytimer 60
```

  **Enable IP routing:**

```
eth ip enable
eth ip addr 192.168.100.1 255.255.255.0
```

  **Define DHCP settings for DNS servers, domain:**

```
dhcp set value domainname flowpoint.com
dhcp set value domainnameserver 192.168.100.68
```

  **Set up DSL parameters:**

```
sd speed 1152
```

  **Define a remote for the Tunnel:**

```
remote add lacclient
remote setpass clientpassword lacclient
remote setLAC tunnelAtHome lacclient
remote setauthen chap lacclient
remote addiproute 192.168.101.0 255.255.255.0 1 lacclient
```

  **Define a remote (internet):**

```
remote add internet
remote setphone isdn 1 5552000 internet
remote setphone isdn 2 5554000 internet
remote setauthen chap internet
remote setpasswd internet internet
remote addiproute 0.0.0.0 0.0.0.0 1 internet
remote setsrcipaddr 192.168.110.1 255.255.255.255 internet
remote addiproute 192.168.101.1 255.255.255.255 1 internet
remote setprotocol ppp internet
remote setpvc 0*38 internet
```

  **Define the actual tunnel:**

```
l2tp add tunnelAtHome
l2tp set chapsecret tunnelsecret tunnelAtHome
l2tp set ourtunnelname tunnelAtWork tunnelAtHome
save
reboot
```

# IPSec (Internet Protocol Security)

**Note:** IPSec security is a software option for your router. It can be purchased as a software option key (see Software Option Keys, page 99). The following section applies only to routers with this option.

IPSec is an open standard that defines optional authentication and encryption methods at the IP packet level. It is a true network layer protocol that provides authentication, privacy, and data integrity. Its protocol suite is comprised of:

- **ESP** (Encapsulated Security Payload)—a security protocol that completely encapsulates and optionally encrypts and/or authenticates user data.
- **AH** (Authentication Header)—a security protocol that authenticates each data packet.
- **IKE** (Internet Key Exchange)—a security protocol used to establish a shared security policy and authenticated keys before an IPSec data transfer begins.

IPSec sessions are initiated through Security Associations (SAs), which allow peers to negotiate a common set of security attributes. In a nutshell, IPSec assures source authenticity, data integrity and confidentiality of IP packets, providing the level of security required by Virtual Private Networks (VPNs).

IPSec can be used in conjunction with L2TP (see L2TP Tunneling — Virtual Dial-Up, page 106). IPSec offers greater security than L2TP, but it does not support as many network protocols. However, bridged and lower layer protocol traffic may be transmitted across an IPSec network if packets are first encapsulated by L2TP, and then by IPSec.

IPSec does not require modification of individual applications or devices for secure data transport. Although it does require global IP addresses for all peers, Network Address Translation (NAT) may be used with IPSec. (See Network Address Translation (NAT), page 91.)

## Transport and Tunnel Encapsulation Modes

IPSec has two encapsulation modes: transport mode and tunnel mode. Transport mode protects traffic between two nodes or peers (the endpoints of the communication). Tunnel mode protects traffic between peers and/or gateways, such as traffic on a VPN or on any other connection where one or both of the endpoints might *not* be IPSec systems.

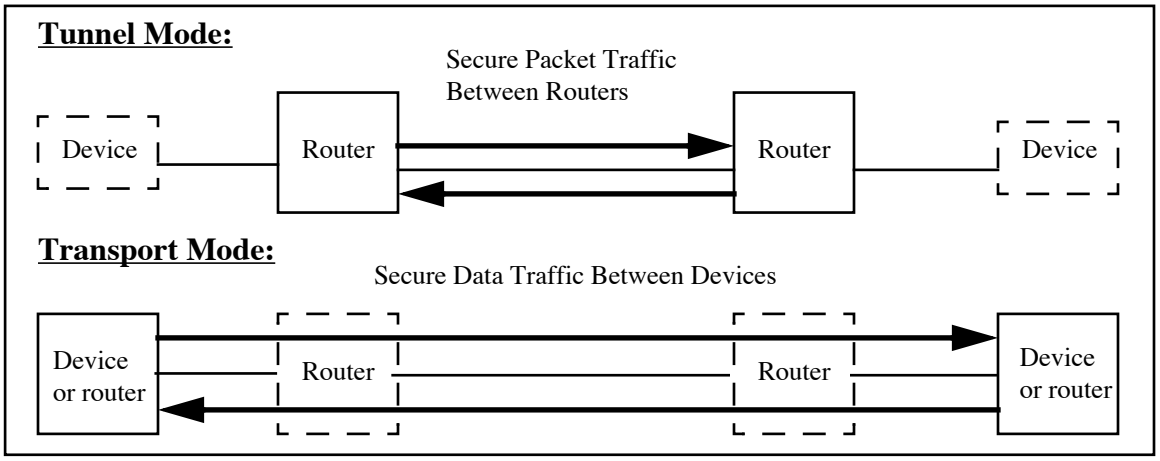
The router supports both IPSec encapsulation methods. It can serve as the endpoint of a *tunnel* mode connection or as the endpoint of a *transport* mode connection. Also, while operating in tunnel mode, the router will allow transport mode traffic to flow through it.

Tunnel mode is the default encapsulation mode for the router. It is used when the IPSec packet comes from either another device or from the encrypting device. In tunnel mode, the IP header is encrypted as part of the payload, and an entirely new IP header is added to the packet. Tunnel mode prevents network traffic analysis. A network attacker could determine the tunnel endpoints (the gateway addresses), but not the true source and destination of the tunneled packets, even if they are the same as the tunnel endpoints

Transport mode is used when the IPSec packet originates in the encrypting device. In transport mode, only the payload (data portion) of each IP packet is encapsulated and/or encrypted. An IPSec header is inserted between the IP header and the upper layer protocol header.

The router should be configured for transport mode when a client is communicating directly with the router. For example, when a remote user wants to access the HTML setup pages or Telnet into the router. It can also be used

for L2TP over IPSec. The routers at either end of the L2TP tunnel do both the IPSec and L2TP encapsulations so the routers can use transport mode for communications.




## ESP and AH Security Protocols

An IPSec connection must use either the AH or the ESP security protocol.The protocol selected determines the encapsulation method used. In addition, if the AH protocol is selected, it can perform packet authentication only, not encryption. If the ESP protocol is selected, it can perform encryption, authentication, or both encryption and authentication.

If ESP encryption is selected, ESP automatically encrypts the data portion (payload) of each packet using the chosen encryption method, DES (56-bit) or 3DES (168-bit).

**Caution:** Restrictions may exist on the export of the DES and 3DES encryption options outside the United States or Canada.

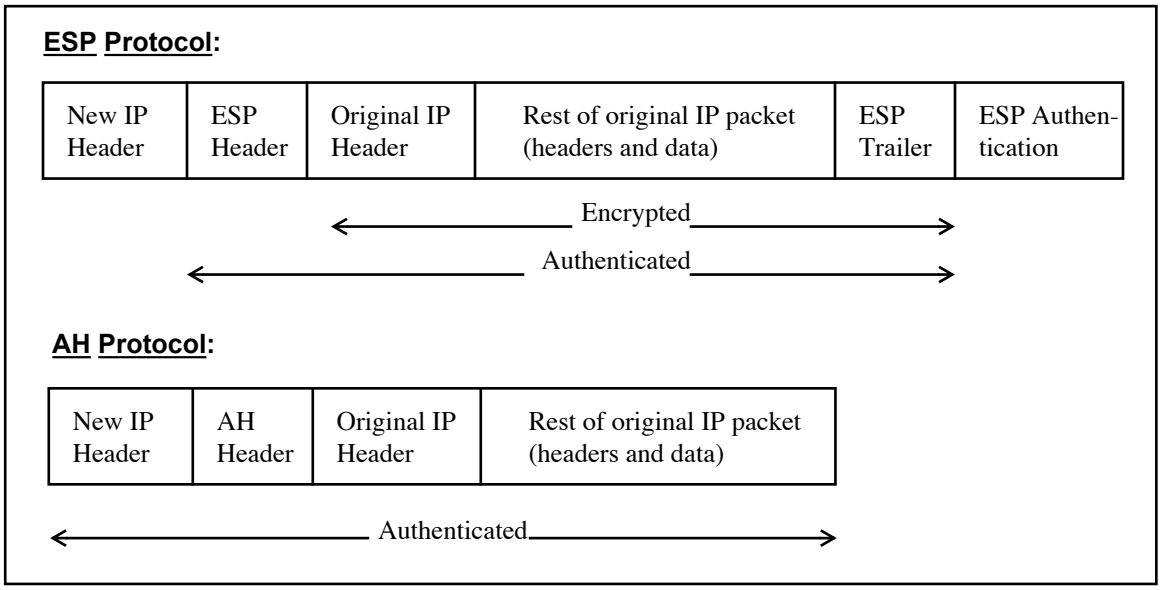
Although encryption cannot be specified for individual applications, a server could be partitioned to achieve the same effect. Given that packets can be encrypted using any combination of security association (SA), protocol, source port, and destination port, you could specify that traffic to and from one database be encrypted while allowing unencrypted traffic to pass freely to and from other databases on the server.

Both the ESP and AH protocols support authentication and replay detection. Replay detection uses sequence numbers to reject old or duplicate packets. The packet is authenticated using a message digest derived from either of two hashing algorithms—SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5).

The ESP protocol can authenticate the data origin and data integrity; it does not authenticate the entire packet. More specifically, the message digest is inserted following, not before, the payload. Both the message digest and payload are sandwiched between the ESP header and ESP trailer.

The AH protocol can perform packet authentication. The AH header protocol defines authentication methods for both the packet's outer IP header and its payload. Unlike ESP authentication, the message digest is inserted in front of the payload.

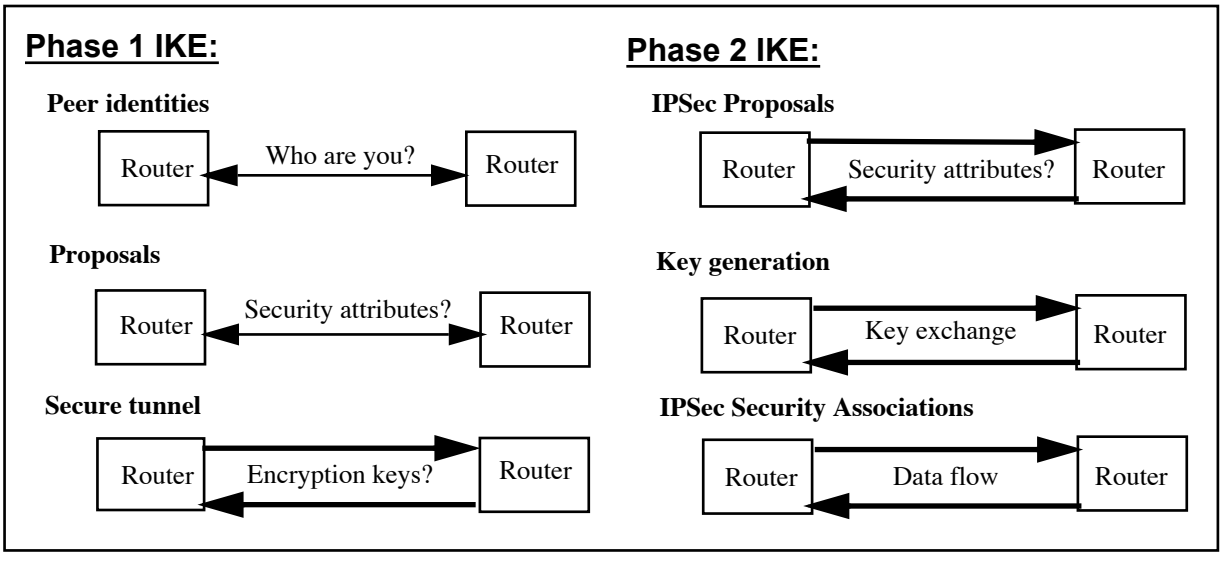
The following figure shows the transformed IP packet after the ESP or AH protocol has been applied in tunnel mode.




## IKE Management

Internet Key Exchange (IKE) management makes encryption key exchange practical, even in large networks where there are many unknown intermediate links between sending and receiving nodes. Unlike protocols that allow only one key exchange per session, IKE can generate and transfer multiple keys between peers during a single tunnel session. Users may specify the duration for which keys are valid. This dynamic type of Diffie-Hellman key exchange greatly reduces the chances of a network attacker finding an entry into a tunnel.

Because VPN users are likely to be using a variety of protocols, a common set of security attributes must be negotiated at the beginning of any tunnel session. Phase 1 IKE is responsible for negotiating these security attributes and establishing peer identities. A secure tunnel for the exchange of encryption keys is also created during this phase. Phase 2 IKE then exchanges proposals for IPSec security attributes, generates the encryption keys and sets up IPSec Security Associations (SAs) for moving user data.

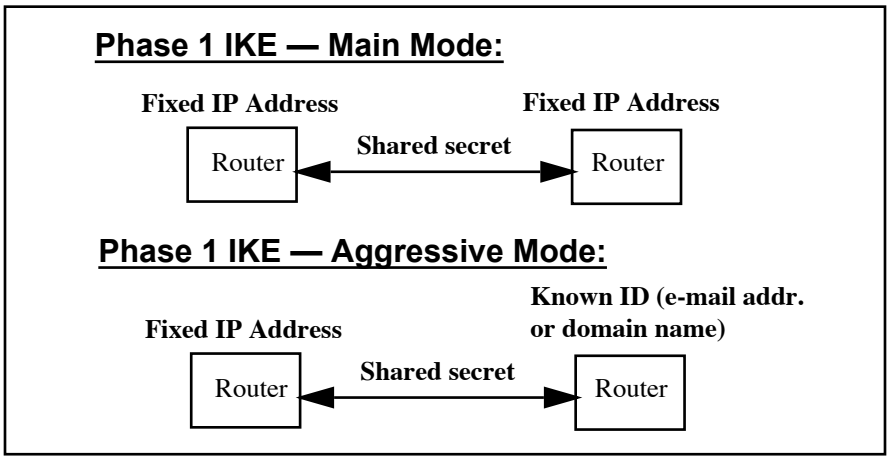


## Main Mode and Aggressive Mode

The router supports two Phase 1 IKE modes: main mode and aggressive mode. These modes apply only to the Phase 1 negotiations, not to the ensuing data transmission.

Main mode is used when both source and destination IP addresses are known. In main mode, only two options require definition initially—the remote peer IP address and the shared secret.

Aggressive mode is used when either the source or destination IP address could change, as with a remote modem or DSL connection. In aggressive mode, additional information must be specified at the beginning of a session. This additional information includes the remote gateway's IP address, the local and remote peer IDs, and an ID type. This information is checked against the router's Security Association (SA) database. If a match is found, a tunnel session can be established.




## Additional IKE Settings

In addition to the peer identification and shared secret described earlier, IKE requires that the router be configured with the following information:

- Session authentication
- Phase 1 IKE message authentication
- Phase 1 IKE message encryption
- One of the following for each IKE proposal:
  —IPSec AH packet authentication
  —IPSec ESP data authentication
  —IPSec ESP data encryption
  —IPSec ESP data authentication *and* data encryption
- Diffie-Hellman key generation group
- IPSec policy (filter) setup
- Policy and peer associations
- Policy and proposal associations

## Security Associations (SAs)

A Security Association (SA) is an instance of security policy and keying material applied to a data flow. Both IKE and IPSec use SAs. An IKE SA is used by IKE only, and unlike IPSec SAs, it is bi-directional. Because it is bi-directional, only one IKE SA is needed for a secure connection.

After an IKE SA is established, any number of IPSec SAs may be created. Although IPSec SAs can be configured manually, most networks rely on IKE to set them up. IKE negotiates and establishes SAs on behalf of IPSec. SAs are negotiated between the two endpoints of the tunnel and contain information on sequence numbering for anti-replay.

IPSec SAs are unidirectional so a set of SAs is needed for a secure connection, one per direction (inbound and outbound) per protocol. If a connection uses both ESP and AH security protocols, IKE needs to establish two SAs for each direction, requiring a total of four SAs for the connection.

IKE negotiates SAs in the following sequence:

**Phase 1 IKE:**

The session initiator creates a cookie and sends it to the responder, with a zero placeholder in the responder cookie area. The responder then creates a cookie and fills in the zeros. All packets will contain these two cookies until the Phase 1 SA expires. IKE Peer commands next establish the identity of local and remote peers. Then IKE Proposal commands specify how packets will be encrypted and/or authenticated for the initial exchange.

**Phase 2 IKE**

IKE IPSec Proposal commands specify *how* packets will be encrypted/authenticated for the final SA. Then IKE IPSec Policy commands specify *which* packets will be encrypted/authenticated for the final SA.

# IKE Commands

The Internet Key Exchange (IKE) process consists of two phases. In phase 1, a moderately secure connection is established between the two security endpoints. This connection is used to exchange key and connection information for the final SA, which is used to exchange user data.

You can use the following command to clear all IKE configuration information from the router.

**ike flush**

The other IKE commands relate to the four categories of information required to set up IKE in the router.

1. The IKE *Peer* commands establish the identity of the local and remote peers.
2. The IKE *Proposal* commands define the proposals exchanged during the Phase 1 exchange.
3. The IKE *IPSec Proposal* commands specify the parameters for the final SA.
4. The IKE *IPSec Policy* commands specify the filtering parameters for the final SA.

## IKE Peer Commands

The IKE peer commands establish the identity of the local and remote peers.

**ike peers add** <*PeerName*> Defines the name of a new IKE peer.

**ike peers delete** <*PeerName*> Deletes an existing IKE peer.

**ike peers list** Lists the IKE peers.

The following commands define the peer connection.

**ike peers set mode** <MAIN | AGGRESSIVE> <*PeerName*>

Sets the peer connection to either main or aggressive mode. Main mode is used when the IP addresses of both ends are known. Aggressive mode is used when the address of one end can change, as with a typical modem or DSL connection.

For a *main mode* connection, set only the IP address and the secret:

**ike peers set address** <*IPaddress*> <*PeerName*>

Sets the IP address of the other endpoint. In a main mode configuration, the other endpoint is constant.

**ike peers set secret** <*secret*> <*PeerName*>

Sets the shared secret for the peer. The secret must be identical for both ends. It can be up to 256 characters long; do not use spaces or non-printable characters.

For an *aggressive mode* connection, you must set the IP address and secret and several more options.

**ike peers set address** <*IPaddress*> <*PeerName*>

Sets the IP address of the other endpoint. One end, the gateway, has a fixed IP address. The other end, the client, has a changing address. When configuring the client, set the peer IP address to the gateway's fixed address. When configuring the gateway for aggressive mode, set the IP address to **0.0.0.0**.

**ike peers set secret** <*secret*> <*PeerName*>

Sets the shared secret for the peer. The secret must be identical for both ends.It can be up to 256 characters long; do not use spaces or non-printable characters.

**ike peers set localid** <*AggressiveModeID*> <*PeerName*>

Sets the local ID. This must match the peer ID on the other end.

**ike peers set localidtype** <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

Sets the type of the local ID (IP address, domain name, or e-mail address). This must match the peer ID type on the other end.

**ike peers set peerid** <*AggressiveModeID*> <*PeerName*>

Sets the peer ID. This must match the local ID on the other end.

**ike peers set peeridtype** <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

**ike peers set peeridtype** <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

Sets the type of the peer ID (IP address, domain name, or e-mail address).This must match the local ID type on the other end.

## IKE Proposal Commands

The IKE proposal commands define the proposals exchanged during the Phase 1 SA.

**ike proposals add** <*ProposalName*> Defines the name of a new IKE proposal.

**ike proposals delete** <*ProposalName*> Deletes an existing IKE proposal.

**ike proposals list** Lists the IKE proposals.

The following commands specify the contents of the proposals exchanged.

**ike proposals set session_auth** <PRESHARE> <*ProposalName*>

Proposes the session authentication; preshared key is currently the only option.

**ike proposals set encryption** <DES | 3DES> <*ProposalName*>

Proposes the encryption method used, as follows:

DES 56-bit encryption

3DES 168-bit encryption

**ike proposals set message_auth** <NONE | MD5 | SHA1> <*ProposalName*>

Proposes the message authentication performed. It can propose no message authentication or authentication using the hash algorithm Message Digest 5 (MD5) or Secure Hash Algorithm-1 (SHA1).

**ike proposals set dh_group** <NONE | 1 | 2 | 5> <*ProposalName*>

Proposes the Diffie-Hellman (DH) key generation group used (no group or group 1, 2, or 5).

**ike proposals set lifetime** <*seconds*> <*ProposalName*>

Proposes the length of time (in seconds) before the Phase 1 SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

## IKE IPSec Proposal Commands

The IKE IPSec proposal commands define the proposals exchanged to set up an IPSec SA, that is, an SA to be used for the user data transfer.

**ike ipsec proposals add** <*ProposalName*> Defines the name of a new IKE IPSec proposal.

**ike ipsec proposals delete** <*ProposalName*> Deletes an existing IKE IPSec proposal.

**ike ipsec proposals list** Lists the IKE IPSec proposals.

The followings *proposals set* commands specify the contents of the proposals exchanged.

**Note:** The following three commands determine the encapsulation method (AH or ESP) used and the authentication and/or encryption performed. You cannot request both AH and ESP encapsulation in the same proposal. You can request any *one* of the following: AH authentication, ESP encryption, ESP authentication, or ESP encryption *and* authentication.

**ike ipsec proposals set espenc** <DES | 3DES | NULL | NONE> <*ProposalName*>

Determines whether ESP encryption is requested and, if it is requested, the encryption method used.

| | |
|---|---|
| DES | Use ESP encapsulation and 56-bit encryption |
| 3DES | Use ESP encapsulation and 168-bit encryption (if 3DES is enabled in the router; see Software Option Keys, page 99.) |
| NULL | No encryption, but use ESP encapsulation. Headers are inserted as though the data was encrypted. This allows verification of the source, but sends the data in the clear, increasing throughput. |
| NONE | No encryption and no ESP encapsulation. (If you select this option, the encapsulation method must be requested by a **set espauth** or **set ahauth** command.) |

**ike ipsec proposals set espauth** <MD5 | SHA1 | NONE> <*ProposalName*>

Determines whether ESP message authentication is requested and, if it is requested, the hash algorithm used.

| | |
|---|---|
| MD5 | Use ESP encapsulation and authenticate using hash algorithm Message Digest 5. |
| SHA1 | Use ESP encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1. |
| NONE | No ESP encapsulation and no ESP message authentication. (If you select this option, the encapsulation method must be requested by a **set espenc** or **set ahauth** command.) |

**ike ipsec proposals set ahauth** <MD5 | SHA1 | NONE> <*ProposalName*>

Determines whether AH message authentication is requested and, if it is requested, the hash algorithm used.
**Note:** The proposal cannot request both AH encapsulation and ESP encapsulation.

| | |
|---|---|
| MD5 | Use AH encapsulation and authenticate using hash algorithm Message Digest 5. |
| SHA1 | Use AH encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1. |
| NONE | No AH encapsulation and no AH message authentication. (If you select this option, the encapsulation method must be requested by a **set espenc** or **set espauth** command.) |

**ike ipsec proposals set ipcomp** <NONE | LZS> <*ProposalName*>

Proposes either no compression or LZS compression.

**ike ipsec proposals set lifetime** <*seconds*> <*ProposalName*>

Proposes the length of time (in seconds) before the IPSec SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

**ike ipsec proposals set lifedata** <*kbytes*> <*ProposalName*>

Proposes the maximum number of kilobytes for the IPSec SA; 0 means unlimited. After the maximum data is transferred, IKE renegotiates the connection. By limiting the amount of data that can be transferred, you reduce the likelihood of the key being broken.

## IKE IPSec Policy Commands

The IKE IPSec policy commands specify the filtering parameters for the IPSec SA.

**ike ipsec policies add** <*PolicyName*> Defines the name of a new IPsec policy.

**ike ipsec policies delete** <*PolicyName*> Deletes an existing IPSec policy.

**ike ipsec policies list** Lists the IPSec policies.

**ike ipsec policies enable** <*PolicyName*> Indicates that the specification of this IPSec policy is complete and enables use of the policy.

**ike ipsec policies disable** <*PolicyName*> Disables an IPSec policy.

The following commands define the filtering parameters for the policy.

**ike ipsec policies set peer** <*PeerName*> <*PolicyName*>

Specifies an IKE peer that may be used for the connection. (The peer must have been defined by IKE peer commands.)

**ike ipsec policies set mode** <TUNNEL | TRANSPORT> <*PolicyName*>

Specifies the encapsulation mode (tunnel or transport) that may be used for the connection. The default is tunnel mode.

**ike ipsec policies set proposal** <*ProposalName*> <*PolicyName*>

Specifies an IKE IPSec proposal that may be used for the connection. (It must have been defined by IKE IPSec proposal commands.) The policy may allow more than one value the proposal parameter. For example, two **set proposal** commands could specify two proposals, either of which could be used by the connection.

**ike ipsec policies set source** <*IPaddress*> <*IPmask*> <*PolicyName*>

Requires the data come from the specified source IP address and mask.

**ike ipsec policies set dest** <*IPaddress*> <*IPmask*> <*PolicyName*>

Requires the data be intended for the specified destination IP address and mask.

**ike ipsec policies set protocol** <*ProtocolNumber* | TCP | UDP | *> <*PolicyName*>

Requires a specific protocol that must be used or allows any protocol (*).

**ike ipsec policies set sourceport** <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

Requires a specific source port for the data or allows any source port (*) (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

**ike ipsec policies set destport** <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

Requires a specific destination port for the data or allows any destination port (*). (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

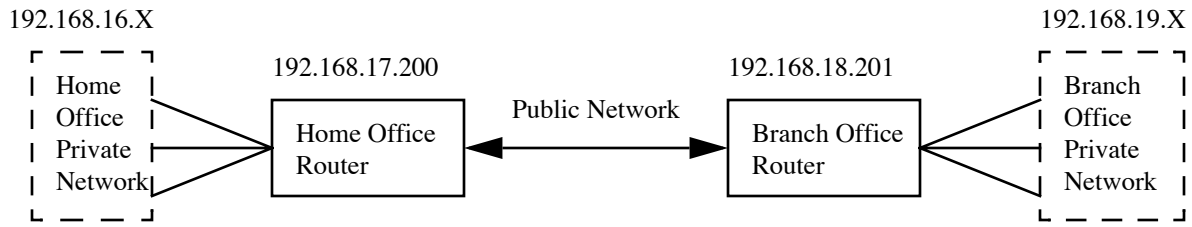
# IKE Configuration Examples

This section shows two simple IKE configurations. The installation CD also contains sample configuration files. These files can be edited for your installation and copied to the router using TFTP or the Windows Quick Start application. For more information on TFTP use, see Batch File Command Execution, page 279.

The first example in this section shows an IKE configuration that uses main mode for a secure connection between two routers with fixed IP addresses. The second example shows how the first configuration must change when one of the routers no longer has a fixed IP address thus, requiring aggressive mode.

## Main Mode Example

The following example lists two setup files that configure two routers for an IKE main mode connection. The two routers are referred to as the home office router and the branch office router.




The configuration sets up a secure connection between the two routers across a public network, thus, the routers are identified by their public IP addresses on the **ike peers** commands. The packets that are transmitted through this secure connection are from devices in the home office and branch office networks. These networks use private addresses, and thus the packets contain private IP addresses. The **ike ipsec policies** commands specify these private source and destination addresses.

This is the file for the home office router:

```
#Home office example using IKE
#Home router private network addresses are 192.168.16.X
#Home router public address is 192.168.17.200
#Branch router private network addresses are 192.168.19.X
#Branch router public address is 192.168.18.201

#Describe the branch office peer
# IKE main mode is used because the branch office has a fixed IP address
# (192.168.18.201). The shared secret is ˝ThisIsASecret12345;)˝
ike peers add branch_peer
ike peers set mode main branch_peer
ike peers set address 192.168.18.201 branch_peer
```

```
ike peers set secret ThisIsASecret12345;) branch_peer

#Describe the branch office IKE phase 1 connection
# DES encryption
# MD5 authentication
# Diffie-Hellman group 2 key exchange
# 24-hour timeout
# Unlimited data
ike proposals add branch_proposal
ike proposals set encryption des branch_proposal
ike proposals set message_auth md5 branch_proposal
ike proposals set dh_group 2 branch_proposal
ike proposals set lifetime 86400 branch_proposal

#Describe the desired IPSec connection
# Triple-DES encryption
# SHA1 authentication
# 30-minute timeout
# Unlimited data
ike ipsec proposals add branch_ipsec_prop
ike ipsec proposals set espenc 3des branch_ipsec_prop
ike ipsec proposals set espauth sha1 branch_ipsec_prop
ike ipsec proposals set lifetime 1800 branch_ipsec_prop
ike ipsec proposals set lifedata 0 branch_ipsec_prop

#Describe the packets to be encrypted
# All packets from network 192.168.19.0/24 to network 192.168.16.0/24
ike ipsec policies add branch_policy
ike ipsec policies set source 192.168.16.0 255.255.255.0 branch_policy
ike ipsec policies set dest 192.168.19.0 255.255.255.0 branch_policy
ike ipsec policies set peer branch_peer branch_policy
ike ipsec policies set proposal branch_ipsec_prop branch_policy

#Enable the IKE connection
ike ipsec policies enable branch_policy

#Save the setup and reboot
save
reboot
```

This is the file for the branch office router:

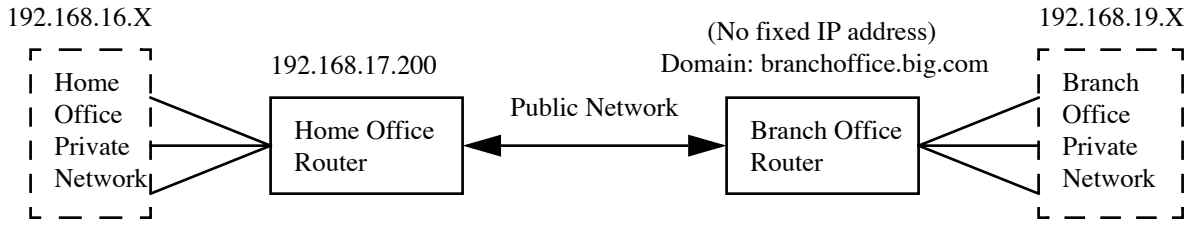
```
#Branch office example using IKE
#Home router private network addresses are 192.168.16.X
```

```
#Home router public address is 192.168.17.200
#Branch router private network addresses are 192.168.19.X
#Branch router public address is 192.168.18.201

#Describe the home office peer
# IKE main mode is used because the home office has a fixed IP address
# (192.168.17.200). The shared secret is ˝ThisIsASecret12345;)˝
ike peers add home_peer
ike peers set mode main home_peer
ike peers set address 192.168.17.200 home_peer
ike peers set secret ThisIsASecret12345;) home_peer

#Describe the home office IKE phase 1 connection
# DES encryption
# MD5 authentication
# Diffie-Hellman group 2 key exchange
# 24-hour timeout
# Unlimited data
ike proposals add home_proposal
ike proposals set encryption des home_proposal
ike proposals set message_auth md5 home_proposal
ike proposals set dh_group 2 home_proposal
ike proposals set lifetime 86400 home_proposal

#Describe the desired IPSec connection
# Triple-DES encryption
# SHA1 authentication
# 30-minute timeout
# Unlimited data
ike ipsec proposals add home_ipsec_prop
ike ipsec proposals set espenc 3des home_ipsec_prop
ike ipsec proposals set espauth sha1 home_ipsec_prop
ike ipsec proposals set lifetime 1800 home_ipsec_prop
ike ipsec proposals set lifedata 0 home_ipsec_prop

#Describe the packets to be encrypted
# All packets from network 192.168.16.0/24 to network 192.168.19.0/24
ike ipsec policies add home_policy
ike ipsec policies set source 192.168.19.0 255.255.255.0 home_policy
ike ipsec policies set dest 192.168.16.0 255.255.255.0 home_policy
ike ipsec policies set peer home_peer home_policy
ike ipsec policies set proposal home_ipsec_prop home_policy
```

```
#Enable the IKE connection
ike ipsec policies enable home_policy

#Save the setup and reboot
save
reboot
```

## Aggressive Mode Example

This example supposes, like the preceding main mode example, that a secure connection is needed between a home office router and a branch office router. However, now the DSL connection for the branch office router does not provide a fixed IP address for the branch office router. Thus, an aggressive mode IKE configuration is required.




To change the main mode configuration to an aggressive mode configuration, you only need to change the **ike peers** commands. All the other IKE commands remain the same. Change the mode to aggressive and change the address of the router that has no fixed address to 0.0.0.0, and specify either its e-mail address or domain name.

**Note:** Remember to save and reboot each router after entering the configuration changes.

Change the **ike peers** commands in the home office router configuration to the following:

```
#Describe the branch office peer
#IKE aggressive mode is required because the branch office does not have
#a fixed IP address. The shared secret is "ThisIsASecret12345;)"
ike peers add branch_peer
ike peers set mode aggressive branch_peer
ike peers set address 0.0.0.0 branch_peer
ike peers set secret ThisIsASecret12345;) branch_peer
ike peers set peeridtype domainname branch_peer
ike peers set peerid branchoffice.big.com branch_peer
ike peers set localidtype ipaddr branch_peer
ike peers set localid 192.168.17.200 branch_peer
```

Change the **ike peers** commands in the branch office router configuration to the following:

```
#Describe the home office peer
#IKE aggressive mode is required because the branch office does not have
#a fixed IP address. The shared secret is "ThisIsASecret12345;)"
ike peers add home_peer
ike peers set mode aggressive home_peer
```

```
ike peers set address 192.168.17.200 home_peer
ike peers set secret ThisIsASecret12345;) home_peer
ike peers set peeridtype ipaddr home_peer
ike peers set peerid 192.168.17.200 home_peer
ike peers set localidtype domainname home_peer
ike peers set localid branchoffice.big.com home_peer
```

# IPSec Commands

The following commands allow you to define an IPSec connection without IKE.

**Note:** If you define a tunnel using IPSec commands, the keys will remain static. This could pose a security risk and is not recommended. Use of IKE for key management is recommended.

| | |
|---|---|
| **ipsec flush** | Clears all IPSec definitions. |
| **ipsec add** <*SAname*> | Defines an SA name. |
| **ipsec del** <*SAname*> | Deletes an existing SA. |
| **ipsec list** [<*SAname*>] | Lists one or all SA entries. |
| **ipsec enable** <*SAname*> | Enables a defined SA entry. |
| **ipsec disable** <*SAname*> | Disables a defined SA entry. |

The following commands define parameters for the specified Security Association (SA).

**ipsec set mode** <TUNNEL | TRANSPORT> <*SAname*>

Requests the encapsulation mode (tunnel or transport) for the SA. The default is tunnel mode.

**ipsec set direction** <INBOUND | OUTBOUND> <*SAname*>

Defines the direction of the SA.

**ipsec set gateway** <*IPaddress*> <*SAname*>

Defines the IP address of the gateway.

**ipsec set encryption** <NULL | DES-CBC | 3DES> <*SAname*>

Selects no encryption, DES (56-bit) encryption or 3DES (168-bit) encryption.

**ipsec set authentication** <SHA1 | MD5> <*SAname*>

Selects authentication using either SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5)

**ipsec set enckey** <*key*> <*SAname*>

Specifies the encryption key (in hexadecimal, 64 bits for DES or 192 bits for 3DES).

**ipsec set authkey** <*key*> <*SAname*>

Specifies the authentication key (hexadecimal).

**ipsec set ident** <*ident*> <*SAname*>

Specifies the identifier (SPID) for the IPSec tunnel. It must match the SPID at the other end of the tunnel, that is, the tx SPID on this end must match the rx SPID on the other end.

**ipsec set service** <ESP | AH | BOTH> <*SAname*>

Selects the authentication and/or encryption services used: AH authentication, ESP encryption, or both ESP encryption and ESP authentication (encryption applied first and then authentication).

**ipsec set compression** <NONE | LZS> <*SAname*>

Selects either LZS compression or no compression.

# Chapter 5. Command Line Interface Reference

This chapter lists the formats of the commands you can enter using the Command Line Interface.

The commands are organized as follows:

- System-level commands
- File system commands
- Frame commands
- Router configuration commands:

  system
  eth
  remote
  adsl
  atm
  eth (specific to the Dual- Ethernet router)
  hdsl
  isdn (specific to the IDSL router)
  sdsl
  dhcp
  l2tp
  filters
  save
  erase
  ike
  ipsec

## Command Line Interface Conventions

The Command Line Interface follows these conventions:

- Command line length may be up to 120 characters long.
- The Command Line Interface is not case-sensitive except for passwords and router names.
- All parameters are positional; i.e., each keyword/parameter must be entered in the correct order, as shown in the command format in this manual.

The command formats shown in this manual follow these conventions:

- Items that appear in **bold** type must be typed exactly as they appear.However, commands can be shortened to just those characters necessary to make the command unique.
- Items that appear in *italics* are placeholders representing specific information that you supply.
- Parameters enclosedin the characters < and > must be entered.
- Parameters enclosed in the characters [ and ] are optional.

Sample command responses are shown in this chapter. In many cases, only the command prompt **#** is returned. If you have not entered the correct parameters, the syntax of the command is displayed.

## ? OR HELP

By entering **?** or **help**, you can list the commands at the current level as well as subcommands. At the lowest subcommand level, entering a **?** may return the syntax of the command. Note that some commands require a character string and the **?** will be taken as the character string if entered in that position.

| **?** or **help** |
|---|

**Example:**

```
# ?
Top-level commands:
?                   help              version
filter              logout            exit
reboot              mem               ps
copy                dir               delete
rename              execute           format
sync                msfs              ifs
ipifs               iproutes          arp
ipxroutes           ipxsaps           bi
system              eth               save
erase               key               remote
call                ping              tcp
dhcp                l2tp              ipsec
ike                 atom              dsp
sdsl                voice
```

# System-Level Commands

These commands are online action and status commands. They allow you to perform the following functions:

- log into and log out of configuration update mode
- display the router's configuration, the version and level numbers
- list running tasks, memory, and communication interfaces
- connect to a remote router to test the line
- list IP routes, IPX routes and SAPs, and root bridge
- save the new configuration image
- reboot the system

## ARP DELETE

Deletes the IP address of the entry in the ARP table.

| **arp delete** <*ipaddr*>|all |
|---|

*ipaddr* IP address in the format of 4 decimals separated by periods.

all Deletes all existing arp table entries

**Example:** `arp delete 128.1.2.0`

## ARP LIST

Lists Address Resolution Protocol (ARP) table entries in an IP routing environment. ARP is a tool used to find the appropriate MAC addresses of devices based on the destination IP addresses.

| **arp list** <*ipaddr*><*InterfaceName*> <*InterfaceUnit*> |
|---|

*ipaddr* IP address associated with a MAC address for a device on the local interface in the format of 4 decimals separated by periods.

*InterfaceName* MAC address on the local network

*InterfaceUnit* For an Ethernet interface, this can be a 1 or 0. For a DSL interface, this is a VPN number.

**Example:** `arp list`

**Response:**

```
IP Addr              Mac Address              Interface
192.84.210.148       00:05:02:00:80:A8        ETHERNET/0
```

## BI

Lists the root bridge, and indicates whether the router is learning, listening, or forwarding.

| **bi** |
|---|

**Response:**

```
# bi
GROUP  0Our  ID=8000+00206f0249fc  Root  ID=8000+00206f0249fc
Port ETHERNET/0              00+00 FORWARDING
```

## BI LIST

Lists MAC addresses and corresponding bridge ports as learned by the bridge function. This list includes several flags and the number of seconds elapsed since the last packet was received by the MAC address.

| **bi list** |
|---|

**Example:**

```
# bi list
BRIDGE GROUP 0:
00206F024C34:                     P     US    SD A
0180C2000000:                     P              A         MC
FFFFFFFFFFFF:                     P FLD          A          BC MC
02206F02E70D: ETHERNET/0      325                  FWD
00C04F2E1AEB: ETHERNET/0      143                  FWD
0060081BD761: ETHERNET/0       95                  FWD
```

Here are the meanings of some of the output flags:

| | |
|---|---|
| FWD | Forward |
| US | This router |
| BC | Broadcast |
| P | Permanent |

## CALL

Dials a remote router. This command can be used to test the ISDN link or L2TP secession and the configuration settings for the remote router.

| **call** <*remoteName*> |
|---|

**Response:**

```
# Request Queued
```

## EXIT

Has the same function as **logout**, but will disconnect you from a Telnet session.

| **exit** |
|---|

## IFS

Lists the communication interfaces installed in the router and the status of the interfaces.

| **ifs** |
|---|

**Response:**

```
# ifs
Interface       Speed       In %      Out %       Protocol      State       Connection
ETHERNET/0      10.0mb      0%/0%     0%/0%       (Ethernet)    OPENED
ATM_VC/1        25.6mb      0%/0%     0%/0%       (CLEAR)       OPENED      to HQ
ATM-25/0        25.6mb      0%/0%     0%/0%       (ATM)         OPENED
CONSOLE/0       9600 b      0%/0%     0%/0%       (TTY)         OPENED
```

## IPIFS

Lists the IP interface.

| ipifs |
|---|

**Response:**

```
ATM_VC/1              192.168.254.1 (FFFFFF00) dest 192.168.254.2 sub 192.168.254.0
                      net 192.168.254.0 (FFFFFF00) P-2-P
ETHERNET/0            192.84.210.12 (FFFFFF00) dest 0.0.0.0 sub 192.84.210.0
                      net 192.84.210.0 (FFFFFF00) BROADCAST
```

## IPROUTES

Lists the current entries in the IP routing table.

| iproutes |
|---|

**Response:**

```
# iproutes
IP route          /  Mask     -->  Gateway    Interface    Hops Flags

0.0.0.0           /ffffffff   -->  0.0.0.0    [none]       0 NW PRIV
192.84.210.0      /ffffff00   -->  0.0.0.0    ETHERNET/0   1 NW FW DIR PERM
192.84.210.12     /ffffffff   -->  0.0.0.0    ETHERNET/0   0 ME
192.168.254.0     /ffffff00   -->  0.0.0.0    [none]       0 NW PRIV
192.168.254.1     /ffffffff   -->  HQ         ATM_VC/1     0 ME
192.168.254.2     /ffffffff   -->  HQ         ATM_VC/1     1 FW DIR PRIV
224.0.0.9         /ffffffff   -->  0.0.0.0    [none]       0 ME
255.255.255.255   /ffffffff   -->  0.0.0.0    [none]       0 NW PERM
```

| Where: | | |
|---|---|---|
| | NW | Network |
| | PERM | Permanent (static) |
| | DOD | Initiate link dial-up |
| | FW | Forward |
| | DIR | Direct |
| | ME | This router |

## IPXROUTES

Lists the current entries in the IPX routing table.

| ipxroutes |
|---|

**Response:**

```
# ipxroutes
Network       Gateway      Interface       Hops  Ticks  Flags
00001001:     HQ           [down]          1       4      STATIC FORWARD DOD
```

```
00000456:      (DIRECT)     ETHERNET/0       0       1       FORWARD
```

| | | |
|---|---|---|
| where: | STATIC | Static route |
| | DOD | Initiate link dial-up |
| | FORWARD | |
| | DIRECT | |

## IPXSAPS

Lists the current services in the IPX SAPs table.

| ipxsaps |
|---|

**Response:**

```
# ipxsaps
Service Name     Type  Node number Network   Skt    Hops
SERV312_FP         4    000000000001:00001001:045    1
```

## LOGOUT

Logs out to reinstate administrative security after you have completed changing the router's configuration.

| logout |
|---|

## MEM

The mem command report the amount of ram installed in the router.

| mem |
|---|

**Response:**

```
# mem
Small buffers used.......18         (7% of 256 used)
Large buffers used.......41         (16% of 256 used)
Buffer descriptors used..59         (7% of 768 used)
Number of waiters s/l....0/0

Table memory allocation statistics:
Sizes  16     32     64     128    256    512    1024   2048
Used   34     18     12     3      8      9      8      7
Free   3      1      4      0      1      1      1      1

Sizes  4096   8192
Used   3      1
Free   1      0
Total in use: 51936, total free: 857368 (8272 + 849096)
```

## MLP SUMMARY

Lists the status of the protocols negotiated for an active remote connection. The following are the most common protocols:

- MLP (Multilink Procedure)
- IPNCP (IP routing Network Protocol)
- CCP (Compression Control Protocol)
- BNCP (Bridging Network Protocol)
- IPXCP (IPX Network Protocol)

Open indicates that the protocol is in ready state.
Stopped means that the protocol is defined, but did not successfully negotiate with the remote end.
No message means that the link is not active.

| **mlp summary** |
|---|

**Example:** `mlp summary`

## PING

This command sends an echo message, available within the TCP/IP protocol suite. The echo message is sent to a remote node and returned in order to test connectivity to the remote node. It is particularly useful for locating connection problems on a network.

A status message is issued for each echo message sent.

You cannot ping your own LAN address; you can ping your own WAN address.

To fit the echo message into one ATM cell in routing mode, set the length of user data down to 0 bytes (**-s 0** or **-l 0**).

| **ping** [**-c** ***count***] [**-i** ***wait***] [**- s** ***size*** (or **-l** ***size***)] [**-I** ***srceaddr***] <*ipaddr*> |
|---|

**-c** *count* Number of packets sent (from 1 to 10). The default is 5 packets.

**-i** *wait* Wait period between packets in seconds (from 1 to 10). The default is 1 second.

**-s** *size* Packet data length in bytes (from 0 to 1648). The default is 56 bytes.

**-l** *size* Same as -s *size*.

**-I** *srcaddr* Source IP address contained in the echo message (4 decimals separated by periods). Use this option to force packets into a tunnel or to force use of the management address as the source address.

*ipaddr* IP address (4 decimals separated by periods).

**Examples:**

The followng command requests 2 echo messages sent 7 seconds apart with a packet length of 34 bytes. The messages are sent to IP address 192.168.254.2.

```
# ping -c 2  -i 7 -s 34 192.168.254.2

ping: reply from 192.168.254.2: bytes=34 (data), time<5 ms

ping: reply from 192.168.254.2: bytes=34 (data), time<5 ms

ping: packets sent 2, packets received 2
```

The following command sends packets with the source IP address 192.168.254.254 to the IP address 192.4.210.122. Default values are used for the other options.

```
ping -I 192.168.254.254 192.4.210.122
```

The following command uses management address 192.168.1.2 as the source address when pinging destination address 192.168.100.100.

```
ping -I 192.168.1.2 192.168.100.100
```

# PS

Lists all of the tasks (processes) running in the system and the status of the tasks.

| ps |
|---|

**Response:**

```
# ps

TID:              NAME     FL  P   BOTTOM    CURRENT SIZE
1:IDLE                     02  7   1208f0    121008   2032
3:MSFS_SYNC                03  6   1224a0    122ba8   2032
4:SYSTEM LOGGER            03  5   122cd0    1233d8   2032
5:LL_PPP                   03  5   126750    126e58   2032
6:NL_IP                    03  5   126fe0    1272e0   1000
7:TL_IP_UDP                03  3   127460    127768   1000|
8:TL_IP_TCP                03  3   1278c0    127fd0   2032
9:IP_RIP                   03  4   128120    128420   1000
10:TELNETD                 03  5   128550    128838   1000
11:DUM                     03  5   12b580    12bc88   2032
12:ATM25                   03  1   12c0a0    12c790   2032
13:SNMPD                   03  5   124b60    125a70   4080
14:BOOTP                   03  5   12e3d0    12e6c0   1000
15:CMD                     01  6   12cba0    12d9f8   4080
```

| | |
|---|---|
| TID: | task ID field |
| NAME: | name of the task |
| FL: | flag field |
| P: | number from 1 to 7 with the highest priority equal to 1 |
| BOTTOM: | address of the task stack |
| CURRENT: | current stack pointer |
| SIZE: | stack size in bytes |

## REBOOT

This command causes a reboot of the system. You must reboot to put your initial configuration or later configuration changes into effect.

**Caution:** If you have not saved your changes, a reboot erases your changes. Remember to enter the **save** command before the **reboot** command.

Reboot is *always* required when the following configuration settings are modified:

- System settings
- Ethernet IP address
- Ethernet IPX network number
- TCP/IP and IPX routing
- Remote router default bridging destination
- TCP/IP route addresses
- IPX routes
- SAPs and bridging

Reboot is also required when you add a new remote entry into the remote database.

Reboot ensures that all file system updates are completed. There is a time lag between the **save** commands and the safe storage of the data in FLASH memory. If the power goes off before the data is stored in memory, the data can be lost. *Always* reboot before powering off the router. Alternatively, you can use the **sync** command.

| **reboot** [default\|factory] |
|---|

default — This option deletes the system configuration file and restores the router to its original defaults (before any configuration was entered).

**Note:** [Default] must be fully *spelled out*.

## SAVE

The **save** command saves the entire router's configuration or parts of it to FLASH memory. The keyword in the command determines what is saved.

**Note:** There is a time lag between the response issued by the **save** command and the time when the data is actually stored in FLASH memory. Issue a **sync** command after a **save** command before powering off the router. This commits the changes to FLASH memory.

| **save all \| keys \| dod \| sys \| eth \| filter \| atom \| sdsl \| dhcp \| l2tp \| ipsec \| IKE** |
|---|

Examples:

| | |
|---|---|
| `save all` | Saves the configuration settings for the system, Ethernet LAN, DSL line, and remote router database into FLASH memory. |
| `save atom` | Saves the ATM configuration settings. |
| `save dhcp` | Saves the DHCP configuration settings into FLASH memory. |
| `save dod` | Saves the current state of the remote router database. |
| `save eth` | Saves the configuration settings for the Ethernet LAN into FLASH memory. |
| `save filter` | Saves the bridging filtering database to FLASH memory. A reboot *must* be executed to load the database for active use. |
| `save sys` | Saves the name, message, and authentication password system settings into FLASH memory. |

## ERASE

The **erase** command can erase the entire router's configuration or parts of it from FLASH memory. The parts that can be erased include:

- System
- Ethernet LAN
- DSL and remote router database settings
- DHCP settings
- Filters

Once you erase part of the configuration, you will need to completely reconfigure that part.

**Note:** All of the following **erase** commands require a **reboot** *without* a **save** command to take effect.

**Note:** There is a time lag between the response issued by the **erase** command and the time that the data is actually deleted from FLASH memory. Issue a **sync** command after an **erase** command before powering off the router. This commits the changes to FLASH memory.

| **erase all \| keys \| dod \| sys \| eth \| filter \| atom \| sdsl \| dhcp \| l2tp \| ipsec \| IKE** |
|---|

Examples:

| | |
|---|---|
| `erase all` | Erases the configuration settings for the system, Ethernet LAN, DSL line, DHCP, and remote router database from FLASH memory. |
| `erase atom` | Erases the ATM configuration settings. |
| `erase dhcp` | Erases the DHCP configuration settings from FLASH memory. |
| `erase dod` | Erases the current state of the remote router database. |
| `erase eth` | Erases the configuration settings for the Ethernet LAN from FLASH memory. |

`erase filter` Erases the current bridging filtering database from FLASH memory.When you issue this command you must reboot (*without* a **save**).

`erase sys` Erases the name, message, and authentication password system settings from FLASH memory..

## TCP STATS

Displays the TCP statistics and open connections.

| tcp stats |
|---|

**Example:** `tcp stats`

## VERS

Displays the software version level, source, software options, and amount of elapsed time that the router has been running.

All software options are listed. If the option has a + prefix, the option was enabled using a key. If the option has a ~ prefix, the option is disabled in this router. For more information, see Software Option Keys, on page 99.

| vers |
|---|

**Example:**

```
# vers
FlowPoint/2025 ATM25 Router
FlowPoint-2000 BOOT/POST V3.0.0 (12-Dec-98 18:10)
Software version 3.0.1 (built Wed Jan 7 13:17:37 PST 1999
18:36:15 PST 1999
Maximum users: unlimited
Options: FRAME RELAY, SDSL, RFC1490, +IP ROUTING, IP FILTERING, WEB, ~L2TP,
 ~ENCRYPT, BRIDGE, IPX, CMMGMT
Up for 0 days 0 hours 20 minutes (started 1/7/1999 at 13:28
```

# File System Commands

The file system commands allow you to perform maintenance and recovery on the router. These commands allow you to:

- Format the file system
- List the contents of the file system
- Copy, rename, and delete files

The router file system is DOS-compatible, and the file system commands are similar to the DOS commands of the same name.

## COPY

Copies a file from the source to the destination. This command allows you to update the router software level or to write configuration files to a TFTP server

Issue a **sync** command after a **copy** command to commit the changes to FLASH memory.

**Caution**: No warning message is issued if you copy over an existing file.

| **copy** <*srcfile*> <*dstfile*> |
|---|

*srcfile* Filename of the source file to be copied.It can be either the name of a local file or a file accessed remotely via a TFTP server.

A local filename is in the format: *name.ext*.

A remotely accessed filename is specified as: **tftp@***serveraddr:filename.ext*. The TFTP server address is optional. If the TFTP server address is not specified, the address used is either the one from which the router booted or the one permanently configured in the boot system.

To force use of a specific source address when copying a file from a TFTP server, use this format: **tftp@***serveraddr***-***sourceaddr***:***filename***.***ext*

*dstfile* Destination filename to which the file is copied.

**Examples:**

The following command copies the file KERNELNW on TFTP server 128.1.210.66 to the local file KERNEL.F2K.

```
# copy tftp@128.1.210.66:kernelnw kernel.f2k
Copying...
421888 bytes copied
```

The following command uses the source address 192.168.1.2 when copying the file KERNELNW on TFTP server 192.168.100.100 to the local file KERNEL.F2K.

```
copy tftp@192.168.100.100-192.168.1.2:kernelnw kernel.f2k
```

## DELETE

Removes a file from the file system.

| **delete** <*filename*> |
|---|

*filename* Name of the file to be deleted. The filename is in the format xxxxxxxx.xxx.

**Example:** `delete kernel.f2k`

**Response:**
`kernel.f2k deleted.`

## DIR

Displays the directory of the file system. The size of each file is listed in bytes.

| **dir** |
|---|

**Example:** `dir`

## EXECUTE

This command loads batch files of configuration commands into the router. This allows for customization and simpler installation of the router. A script file can contain commands, comments (lines introduced by the # or ; characters), and blank lines.

There are two kinds of script files:

- A one-time script that is executed on startup (only once).
- A group of commands that can be executed at any time from the Command Line Interface with the **execute <***filename***>** command.

One-time scripts are useful to execute the complete configuration process from a default (unconfigured) state.

| **execute** <*filename*> |
|---|

*filename* Name of the file to be executed.

**Example:** `execute script1`

## FORMAT DISK

Erases and reformats the router file system. This command should *only* be used when the file system is unusable. If the router does not execute the POST test and software boot successfully, and the result of the **dir** command

indicates the file system is corrupted, you may wish to reformat the disk, reboot the router, and recopy the router software.

| **format disk** |
|---|

**Example:** `format disk`

**Response:**

```
NEWFS: erasing disk...
NEWFS: fs is 381k and will have 762 sectors
NEWFS: 128 directory slots in 8 sectors
NEWFS: 747 fat entries in 3 sectors
NEWFS: writing boot block...done.
NEWFS: writing fat tables...done.
NEWFS: writing directory...done.
Filesystem formatted!
```

## MSFS

Checks the structure of the file system. This command performs a function similar to the DOS **chkdsk** command. The router analyzes the File Allocation Table (FAT) and produces a file system status report.

**Warning**: When you specify **fix**, make sure that no other operation is being performed on the configuration files at the same time by Configuration Manager or by another user.

| **msfs** [fix] |
|---|

fix — If fix is specified, errors are corrected in the FAT. This option should *only* be used when an **msfs** command results in a recommendation to apply the **fix** option.

**Example:** `msfs`

**Response:**

```
Filesystem 0, size=825k:
Checking filesystem...
Checking file entries...
       SYSTEM     CNF ... 2304      bytes .. ok.
       ATM25      DAT ... 20        bytes .. ok.
       DHCP       DAT ... 1536      bytes .. ok.
       KERNEL     F2K ... 257014    bytes .. ok.
       IDL_7      AIC ... 14828     bytes .. ok.
       ASIC       AIC ... 14828     bytes .. ok.
       FILTER     DAT ... 1284      bytes .. ok.
       1097 fat(s) used, 0 fat(s) unused, 0 fat(s) unref, 534 fat(s) free
       561664 bytes used by files, 9728 bytes by tables, 273408 bytes free
```

## RENAME

Renames a file in the file system.

| **rename** <*oldName*> <*newName*> |
|---|

*oldName* Existing name of the file. The filename is in the format xxxxxxxx.xxx.

*newName* New name of the file. The filename is in the format xxxxxxxx.xxx.

**Example:** `rename ether.dat oldeth.dat`

**Response:**

```
'ether.dat' renamed to 'oldeth.dat'
```

## SYNC

Commits the changes made to the file system to FLASH memory.

| **sync** |
|---|

**Example:** `sync`

**Response:**

```
Syncing  file systems...done.
```

**Warning**: Syncing is not complete until you see the message "done".

# Frame Commands

## FRAME ?

Lists the frame commands.

| **frame ?** |
|---|

Example:

```
# frame ?
Frame Commands:
?                   help                lmi
stats               voice
```

## FRAME LMI

Turns frame LMI either on or off.

| **frame < on \| off>** |
|---|

Example:

```
# frame on
LMI is on
```

## FRAME VOICE

Displays the voice DLCI for voice routers.

| **frame voice** |
|---|

Example:

```
# frame voice
Voice DLCI is 22
```

## FRAME STATS

Displays frame relay statistics.

| **frame stats** |
|---|

Example:

```
# frame stats
FR/0 Frame Relay Statistics
ANSI LMI:
Protocol Errors..................... 0
Unknown Msg Recv.................... 0
T391 Timeouts....................... 0
PVC Status Changes.................. 0
StatusEnq Sent...................... 0
Status Recv......................... 0
StatusEnq Recv...................... 0
Unconfigured DLCIs recv in Status Msgs. 0

    LMI Stats for DLCI.............. 22
    LMI State....................... UNKNOWN
    Status State Changes............ 0
    Active to Not Active Changes.... 0
    Not Active to Active Changes.... 0
    Data Packets In................. 0
```

```
Data Packets Out.................. 0
Data Packets Out Queued........... 0
Data Packets Out (dropped Q Full).. 0
Voice Cells In.................... 0
Voice Cells In (with errors)...... 0
Voice Cells Out................... 0

LMI Stats for DLCI................ 16
LMI State......................... UNKNOWN
Status State Changes.............. 0
Active to Not Active Changes...... 0
Not Active to Active Changes...... 0
Data Packets In................... 0
Data Packets Out.................. 0
Data Packets Out Queued........... 0
Data Packets Out (dropped Q Full).. 0
Voice Cells In.................... 0
Voice Cells In (with errors)...... 0
Voice Cells Out................... 0

Data Out (Delayed by Voice)....... 0
```

# Router Configuration Commands

Configuration commands are used to set configuration information for each functional capability of the router. Each functional capability has a specific prefix for its associated commands:

- **? or help** Summary of available commands
- **system** Target router system commands
- **eth ip** Ethernet IP routing commands
- **remote** Remote router database commands
- **adsl** Asymmetric Digital Subscriber Line commands (ADSL routers only)
- **atm** Asynchronous Transfer Mode commands (ATM routers only)
- **eth** Dual-Ethernet router commands only
- **hdsl** High-speed Digital Subscriber Line commands (HDSL routers only)
- **isdn (for IDSL)** ISDN Digital Subscriber Line (IDSL routers only)
- **sdsl** Symmetric Digital Subscriber Line commands (SDSL routers only)
- **dhcp** Dynamic Host Configuration Protocol commands
- **l2tp** Layer-2 Tunneling Protocol commands
- **save** Save configuration to FLASH memory commands
- **erase** Erase configuration from FLASH memory commands
- **filter** Filtering commands
- **ike** Internet Key Exchange commands
- **ipsec** IPSec commands

# Target Router System Configuration Commands (SYSTEM)

The following commands set basic router configuration information:

- name of the router
- optional system message
- authentication password
- security authentication protocol
- management security
- system administration password
- IP address translation
- NAT configuration
- host mapping
- WAN-to-WAN forwarding
- filters

## SYSTEM ?

Lists the supported keywords.

| system ? |
|---|

**Example:**

```
# sys ?
System commands:
?                   msg                 name
passwd              authen              community
list                admin               history
log                 addHostMapping      delHostMapping
addServer           delServer           bootpServer
supportTrace        telnetport          snmport
httpport            syslogport          addTelnetFilter
delTelnetFilter     addSNMPFilter       delSNMPFilter
addHTTPFilter       delHTTPFilter       addSyslogFilter
delSyslogFilter     wan2wanforwarding   OneWANdialup
blockNetBIOSDefault addUDPrelay         delUDPrelay
securityTimer       addIPRoutingTable   delIPRoutingTable
moveIPRoutingTable
```

## SYSTEM ADDHOSTMAPPING

This command is used to remap a range of local-LAN IP addresses to a range of public IP addresses on a *system-wide basis*. These local addresses are mapped one-to-one to the public addresses.

**Note:** The range of public IP addresses is defined by *<first public addr>* only. The rest of the range is computed automatically (from *<first public addr>* to *<first public addr>* + number of addresses remapped - 1) inclusive.

| **system addHostMapping** *<first private addr> <second private addr> <first public addr>* |
|---|

| | |
|---|---|
| *first private addr* | First IP address in the range of IP addresses to be remapped, in the format of 4 decimals separated by periods. |
| *second private addr* | Last address in the range of IP addresses to be remapped, in the format of 4 decimals separated by periods. |
| *first public addr* | Defines the range of public IP addresses, in the format of 4 decimals separated by periods.<br>The rest of the range is computed automatically. |
| **Example:** | `system addHostMapping 192.168.207.40 192.168.207.49 10.1.1.7` |

## SYSTEM ADDHTTPFILTER

This command is used to allow devices within the defined IP address range to use the HTTP protocol (for example, to browse the Web). This command is useful to block devices on the WAN from accessing the Web browser.

| **system addHTTPFilter** *<first ip addr>* [*<last ip addr>*] \| LAN |
|---|

| | |
|---|---|
| *first ip addr* | First IP address of the range. |
| *last ip addr* | Last IP address of the range. May be omitted if the range contains only one IP address. |
| LAN | Local Ethernet LAN. |
| **Example:** | `system addHTTPFilter 192.168.1.5 192.168.1.12` |

## SYSTEM ADDIPROUTINGTABLE

Defines a new virtual routing table. Once defined, you can add routes to the table using the commands **eth ip bindRoute** (page 169) and **remote bindIPVirtualRoute** (page 186).

The command specifies the name of the new routing table and the range of IP addresses that reference the table for their routing. When the router receives a packet, the source address of the packet determines which routing table is used. For example, if the range of addresses for the virtual routing table ROSA includes address 192.168.25.25, then every packet with the source address 192.168.25.25 is routed using virtual routing table ROSA.

If the source address of a packet is not within the address ranges for any virtual routing table, the default routing table is referenced to route the packet.

For more information, see Virtual Routing Tables, on page 77.

**system addIPRoutingTable** <*first ip addr*> [<*last ip addr*>] <*tablename*>

*first ip addr* First IP address of the range (4 decimals separated by periods).

*last ip addr* Last IP address of the range (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address. The specified address range may not overlap the address range defined for any other virtual routing table.

*tablename* Name of the virtual routing table to which the addresses are assigned (character string).

**Example:**

The following command defines a virtual routing table named ROSA (if it does not already exist) and assigns it the IP address range 192.168.1.5 through 192.168.1.12.

```
system addIPRoutingTable 192.168.1.5 192.168.1.12 ROSA
```

## SYSTEM ADDSERVER

This Network Address Translation (NAT) command is used to configure a local IP address as the selected server on the LAN (FTP, SMTP, etc.) for the global configuration.

**system addServer <***ipaddr***>**|discard|me <*protocolid*>**>** |tcp|udp <*first port*> |ftp|telnet|smtp|snmp|http [<*last port*> [<*first private port*>]]

*ipaddr* IP address of the host selected as server in the format of 4 decimals separated by periods.

discard Used to discard the incoming server request.

me Used to send the incoming server request to the local router, regardless of its IP address.

*protocolid* Protocol used by the selected server; can be **tcp** or **udp,** or a numeric value.

*first port* First or only port as seen by the remote end**.** Port used by the selected server; can be a string such as **ftp, telnet, smtp, snmp,** or **http**, or a numeric value between 0 and 65,535. A numeric value of 0 will match any port.

*last port* If specified, this is used with <*first port*> to denote a range of ports as seen by the remote end for the server on the LAN.

*first private port* If specified, this is a port remapping of the incoming request from the remote end.

**Example:** `system addServer 192.168.1.5 tcp smtp`

## SYSTEM ADDSNMPFILTER

This command is used to validate SNMP clients by defining a range of IP addresses that are allowed to access the router via SNMP. This validation feature is *off* by default.

**Note 1:** This command does *not* require a reboot and is effective immediately.

**Note 2:** To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password).

| **system addSNMPFilter** <*first ip addr*> [<*last ip addr*>] | LAN |
|---|---|

*first ip addr* First IP address of the client range.

*last ip addr* Last IP address of the client range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

**Example:** `system addSNMPFilter 192.168.1.5 192.168.1.12`

## SYSTEM ADDSYSLOGFILTER

Access to the system logging port can be controlled with the following filter command:

| **system addSyslogFilter <***firstipaddr***>** [<*last ip addr*>] | LAN |
|---|---|

## SYSTEM ADDTELNETFILTER

This command is used to validate Telnet clients by defining a range of IP addresses that are allowed to access the router via Telnet. This validation feature is *off* by default.

**Note 1:** This command does *not* require a reboot and is effective immediately.

**Note 2:** To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (log in with password).

| **system addTelnetFilter** <*first ip addr*> [<*last ip addr*>] | LAN |
|---|---|

*first ip addr* First IP address of the client range.

*last ip addr* Last IP address of the client range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

**Example:** `system addTelnetFilter 192.168.1.5 192.168.1.12`

## SYSTEM ADDUDPRELAY

This command is used to create a UDP port range for packet forwarding. You can specify a port range from 0 to 65535; however, 137 to 139 are reserved for NetBIOS ports. Overlap of UDP ports is not allowed.

| **system addUDPrelay** <*ipaddr*> <*first port*>|all [<*last port*>] |
|---|

| | |
|---|---|
| *ipaddr* | IP address of the server to which the UDP packet will be forwarded. |
| *first port* | First port in the UDP port range to be created. |
| all | Incorporates all the available UDP ports in the new range. |
| *last port* | Last port in the UDP port range to be created. |

**Example:** `system addUDPrelay 192.168.1.5 all`

## SYSTEM ADMIN

Sets the administration password that is used to control write access to the target router configuration.

| **system admin** <*password*> |
|---|

| | |
|---|---|
| *password* | Write-enable login password**.** |

**Example:** `system admin adx1lp`

## SYSTEM AUTHEN

Forces the target router authentication protocol that is used for security negotiation with the remote routers when the local side authentication is set. You should not need to issue this command as the best security possible is provided with the **none** default.

| **system authen** none | pap | chap |
|---|

| | |
|---|---|
| none | When set to none (the default), the authentication protocol is negotiated, with the *minimum* best security level as defined for each remote router in the database. |
| pap | When set to pap, negotiation will begin with PAP (instead of CHAP) for those entries that have PAP in the remote database and only when the call is initiated locally. |
| chap | Overrides all the remote database entries with chap; i.e., only CHAP will be performed. |

**Example:** `system authen CHAP`

## SYSTEM BLOCKNETBIOS

The router can block all netbios and netbui requests from being sent over the wan. This command sets the defaultvalue for the entire router when a remote router is defined.

| **system blockNetBIOS** Default yes\|no |
|---|

After a remote device is defined, the command remote blockNetBIOS on\|off <remoteName> can enable or disable this feature.

## SYSTEM BOOTPSERVER

Lets the router relay BootP or DHCP requests to a DHCP server on the WAN when a PC attempts to acquire an IP address using DHCP. This command disables the router's DHCP server.

| **system bootpServer** <*ipaddr*> |
|---|

*ipaddr* IP address of the target router in the format of 4 decimals separated by periods.

**Example:** `system bootpServer 128.1.210.64`

## SYSTEM COMMUNITY

Enhances SNMP security by allowing the user to change the SNMP community name from its default value of "public" to a different value. Refer to Controlling Remote Management, on page 97.

**Note:** The command **system community** (with no value) will display the current community name.

| **system community** [<*SNMP community name*>] |
|---|

*SNMP community name* String of up to 40 characters.

**Example 1:** `system community fred`

**Example 2:** `system community`

## SYSTEM DELHOSTMAPPING

Undoes an IP address/host translation (remapping) range that was previously established with the command **remote addHostMapping** on a *per-systemwide basis*.

| **system delHostMapping** <*first private addr*> <*second private addr*> <*first public addr*> |
|---|

*first private addr* First IP address in the range of IP address, in the format of 4 decimals separated by periods.

*second private addr* Last address in the range of IP address, in the format of 4 decimals separated by periods.

*first public addr* Defines the range of public IP addresses, in the format of 4 decimals separated by periods.

The rest of the range is computed automatically.

**Example:** `system delHostMapping 192.168.207.40 192.168.207.49 10.1.1.7`

## SYSTEM DELHTTPFILTER

Deletes an IP address range created by the **system addHTTPFilter** command.

**system delHTTPFilter** <*first ip addr*> [<*last ip addr*>] | LAN

*first ip addr* First IP address of the range.

*last ip addr* Last IP address of the range. May be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

**Example:** `system delHTTPFilter 192.168.1.5 192.168.1.12`

## SYSTEM DELIPROUTINGTABLE

Deletes a range of addresses that reference a virtual routing table or deletes the entire virtual routing table.

To list the virtual routing tables, use the **iproutes** command (page 139).

For more information, see Virtual Routing Tables, on page 77.

**system delIPRoutingTable** ALL | <*first ip addr*> [<*last ip addr*>] <*tablename*>

ALL Deletes the virtual routing table. Both the table definition and all routes in the table are deleted.

*first ip addr* First IP address of the range to be deleted (4 decimals separated by periods).

*last ip addr* Last IP address of the range to be deleted (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address.

*tablename* Name of the virtual routing table (character string).

**Examples:**

Deletes two IP addresses from the address range that references routing table ROSA:

```
system delIPRoutingTable 192.168.1.5 192.168.1.6 ROSA
```

Deletes the virtual routing table ROSA:

```
system delIPRoutingTable all ROSA
```

## SYSTEM DELSERVER

Is a Network Address Translation (NAT) command that can be used to delete an entry created by the **system addServer** command.

| **system delServer <***ipaddr***>**\| discard\|me <*protocolid***>** \|tcp\|udp <*first port*> \|ftp\|telnet\|smtp\|snmp\|http [<*last port*> [<*first private port*>]] <*remoteName*> |
|---|

| | |
|---|---|
| *ipaddr* | IP address of the host selected as server in the format of 4 decimals separated by periods |
| discard | Used to discard the incoming server request. |
| me | Used to send the incoming server request to the local router, regardless of its IP address. |
| *protocolid* | Protocol used by the selected server; can be **tcp** or **udp.** |
| *first port* | First or only port as seen by the remote end**.** Port used by the selected server.<br>Can be a string such as **ftp, telnet, smtp, snmp**, or **http**, or a numeric value between 0 and 65,535.<br>A numeric value of 0 will match any port |
| *last port* | If specified, this is used with <*first port*> to specify a range of ports as seen by the remote end for the server on the LAN. |
| *first private port* | If specified, this is a port remapping of the incoming request from the remote end. |
| **Example:** | `system delServer 192.168.1.5 tcp smtp` |

## SYSTEM DELSNMPFILTER

Deletes the client range previously defined by the command **system addsnmpfilter**.

**Note 1:** This command does *not* require a reboot and is effective immediately.

**Note 2:** To list the range of allowed clients, use the command **system list** when you are logged in with read and write permission (be sure to log in with password).

| **system delSNMPFilter** <*first ip addr*> [<*last ip addr*>] \| LAN |
|---|

| | |
|---|---|
| *first ip addr* | First IP address of the client range. |
| *last ip addr* | Last IP address of the client range*;* may be omitted if the range contains only one IP address. |
| LAN | Local Ethernet LAN. |
| **Example:** | `system delsnmpfilter 192.168.1.5 192.168.1.12` |

## SYSTEM DELTELNETFILTER

Deletes the client range previously defined by the command **system addTelnetFilter**.

**Note 1:** This command does *not* require a reboot and is effective immediately.

**Note 2:** To list the range of allowed clients, use the command system list when logged in with read and write permission (be sure to log in with password).

| **system delTelnetFilter** <*first ip addr*> [<*last ip addr*>] | LAN |
|---|---|

*first ip addr* First IP address in the client range.

*last ip addr* Last IP address in the client range; may be omitted if the range contains only one IP address.

LAN Local Ethernet LAN.

**Example:** `system deltelnetfilter 192.168.1.5 192.168.1.12`

## SYSTEM DELUDPRELAY

Deletes the port range that was previously enabled by the command **system addUDPrelay.**

| **system delUDPrelay** <*ipaddr*> <*first port*>| *all [<last port>]* |
|---|

*ipaddr* IP address of the server.

*first port* First port in the UDP port range to be deleted.

*all* Deletes all existing UDP ports.

*last port* Last port in the UDP port range to be deleted.

**Example:** `system delUDPrelay 192.168.1.5 all`

## SYSTEM DELSYSLOGFILTER LAN

This command deletes the specified Syslog filter.

| **system delSyslogFilter <***firstipaddr***> <***lastipaddr***>** |
|---|

## SYSTEM HISTORY

Displays the router's most recent console log.

| **system history** |
|---|

**Example:** system history

## SYSTEM LIST

Lists the target router's system name, security authentication protocol, callerID and data-as-voice status, and system message.

| **system list** |
|---|

**Example:** `system list`

**Response:**

```
GENERAL INFORMATION FOR <SOHO>

System started on..................... 1/7/1998 at 13:29
  Authentication override.......... NONE
WAN to WAN Forwarding.............. yes
  BOOTP/DHCP Server address........ none
  Telnet Port...................... default (23)
SNMP Port............................ default (161)
System message: Configured January 1998
```

## SYSTEM LOG

Allows logging of the router's activity in a Telnet session.

| **system log** start \| stop \| status |
|---|

start — Used to monitor router activity at all times.

**Example:** `system log start`

stop — Used to discontinue the logging utility at the console.

**Example:** `system log stop`

status — Used to find out if other users (yourself included) are using this utility.

**Example:** `system log status`

## SYSTEM MSG

Sets a message that is saved in the target router you are configuring.

| **system msg** <*message*> |
|---|

*message* — Message (character string. Space characters are not allowed within the message; you may use underscore characters instead. If you do not enter a message, the current message is displayed. The message may be no more than 255 characters.

**Example:** `system msg Configured _on_ 10/21/98`

## SYSTEM MOVEIPROUTINGTABLE

Moves a range of IP addresses to another virtual routing table. The command first looks at the address ranges defined for other virtual routing tables, searching for the addresses to be moved. If it finds addresses to be moved, it deletes them from the address ranges for the other virtual routing tables. The command then adds the specified address range to the virtual routing table named on the command.

To list the routes in the virtual routing tables, use the **iproutes** command (page 139) or the **remote listiproutes** command (page 196).

For more information, see Virtual Routing Tables, on page 77.

| **system moveIPRoutingTable** <*first ip addr*> [<*last ip addr*>] <*tablename*> |
|---|

*first ip addr* — First IP address of the range to be moved (4 decimals separated by periods).

*last ip addr* — Last IP address of the range to be moved (4 decimals separated by periods). This parameter may be omitted if the range contains only one IP address.

*tablename* — Name of the virtual routing table to be assigned the address range (character string). The virtual routing table may be new or it may already exist.

**Example:**

Suppose you want all packets with source addresses in the range 192.168.254.11 through 192.168.254.20 to be routed using virtual routing table MIGUEL. Addresses in that range may already be assigned to other virtual routing tables. Therefore, to delete the addresses from any other virtual routing tables and assign the address range to MIGUEL, you enter this command:

```
system moveIPRoutingTable 192.168.254.11 192.168.254.20 MIGUEL
```

## SYSTEM NAME

Sets the name for the target router that you are configuring. You must assign a name to the target router. This name is sent to a remote router during PAP/CHAP authentication.

| **system name** <*name*> |
|---|

*name* — Name of the target router (character string). Space characters are not allowed within the name; you may use underscore characters instead. (The system name is a “word” when exchanged with PAP/CHAP.) If you do not enter a name, the current name of the router is displayed. If you type anything after **system name**, the characters will be taken as the new name.

**Note:** The system name is case sensitive and may be no more than 50 characters.

**Example:** `system name Router1`

## SYSTEM ONEWANDIALUP

This command is useful when security concerns dictate than the router have only one connection active at a time. For example, the command can prevent from connecting to the Internet and to another location such as your company at the same time. The command **system oneWANdialup** on forces the router to have no more than *one* connection to a remote entry active at one time. (Multiple links to the same remote are allowed.)

A connection is only generated when data is forwarded to the remote router (dial-on-demand); Permanent links cannot be automatically generated.

At system startup time, the router examines each remote entry. If if finds one remote enabled, it leaves the remote enabled. If it finds more than one remote entry enabled, then it disables every entry that does not have a protocol of PPP or PPPLLC. It sets the minimum number of active links (**remote minLink**) to 0 (zero) on the enabled entries; if the command did not perform this function, connections to multiple destinations would not be possible (since the link to the destination with **minLink=**non-zero would be active).

The command allows multiple connections to the SAME location and supports PPP Mult-link protocol.

This **system oneWANdialup** command complements the **system** command that controls WAN-to-WAN forwarding. That command allows multiple connections to different locations to be active at the same time but stops traffic from passing from one WAN connection to another.

| **system oneWANdialup** onloff |
|---|

on — Enables only one active connection at a time to a remote entry.

off — Disables **system oneWANdialup.**

**Example:** `system oneWANdialup on`

## SYSTEM PASSWD

Sets the system authentication password for the target router that is used when the router connects to other routers or is challenged by them. This password is a default password used for all remote sites unless a unique password is explicitly defined for connecting to a remote router with the **remote setOurPasswd** command.

| **system passwd** <*password*> |
|---|

*password* — Authentication password of the target router.

**Note:** The password is case-sensitive and should be no more than 40 characters.

**Example:** `system passwd chwgn1`

## SYSTEM SECURITYTIMER

Automatically logs out a Telnet or console user out of privileged mode when no typing has occurred for 10 minutes. This command allows the user to change the 10-minute default to a different value.

| **system securityTimer** <*minutes*> |
|---|

*minutes* Length of time in minutes.
Auto logout can be disabled by setting the <*minutes*> to zero.

**Example:** `system securityTimer 15`

## SYSTEM SNMPPORT

Manages SNMP port access including disabling SNMP, reestablishing SNMP services, or redefining the SNMP port for security reasons. Refer to Chapter 4. Controlling Remote Management on page 97.

**Note:** This command requires a **save** and **reboot** to take effect.

| **system snmpport** default\|disabled \| <*port*> |
|---|

default Restores the default values to 161.

disabled Disables remote management.

*port* Used to define a new SNMP port number. Use this option to redefine the SNMP port to a non- well-known value to restrict remote access.

**Examples:**
```
system snmpport default

system snmpport disabled

system snmpport 3333
```

## SYSTEM SUPPORTTRACE

Lets you capture to a file all the configuration data that Technical Support may need to investigate configuration problems. This exhaustive list command incorporates the following commands:

- s**ystem history**
- **vers**
- **mem**
- **system list**
- **eth list**
- **dhcp list (if DHCP is enabled)**
- **remote list**
- **ifs**
- **bi (if bridging is enabled)**
- **ipifs**
- **iproutes**
- **ipxroutes**

| system supporttrace |
|---|

**Example:** `system supporttrace`

## SYSTEM TELNETPORT

The router has a built-in Telnet server. This command is used to specify which router's TCP port is to receive a Telnet connection.

**Note:** This command requires a **save** and **reboot** to take effect.

| **system telnetport** default\|disabled\|<*port*> |
|---|

default — The default value is 23.

disabled — The router will not accept any incoming TCP request.

*port* — Port number of the Ethernet LAN. It is recommended that this number be > 2048 if not 0 (disabled) or 23 (default).

**Examples:**

```
system telnetport default

system telnetport disabled

system telnetport 3333`
```

## SYSTEM WAN2WANFORWARDING

Allows the user to manage WAN-to-WAN forwarding of data from one WAN link to another.

For example, an employee uses the router at home to access both a company network and the Internet at the same time, and the company does not want its information to pass to the Internet, then this command is useful for disabling WAN-to-WAN forwarding.

| **system wan2wanforwarding** on\|off |
|---|

on — Allows data to be forwarded from one WAN link to another WAN link.

off — Stops data from being forwarded from one WAN link to another WAN link.

**Example:** `system wan2wanforwarding on`

# Target Router Ethernet LAN Bridging and Routing (ETH)

The following commands allow you to configure the Ethernet interfaces in your router. You can:

- Set the Ethernet LAN IP address
- Define logical interfaces to provide service to multiple IP subnets
- Manage the contents of the default routing table and any virtual routing tables
- Enable and disable IP routing
- List the current configuration settings

**Note:** In general, these commands require a **save** and **reboot** before they take effect. However, changes made to IP filters and to virtual routing tables take effect immediately; the changes are lost, though, if they are not saved before the next **reboot**.

## ETH ?

Lists the supported keywords.

| **eth ?** |
|---|

**Example:**

```
# eth ?
Ethernet commands:

?            add            delete
br           ip             ipx
list         mtu

# eth ip ?
eth ip sub-commands
?            addr           ripmulticast
options      enable         disable
firewall     directedBcast  addroute
delroute     defgateway     bindRoute
unbindRoute  filter
```

## ETH ADD

Adds a logical interface onto an Ethernet port so that the router can provide service to multiple IP subnets. The **eth add** command defines the port number and logical interface number. You should then use an **eth ip addr** command to define the IP subnet that uses the logical interface. For more information, see Multiple IP Subnets, on page 76.

A logical interface 0 always exists for Ethernet port 0 (and for port 1 in a dual-port router); logical interface 0 cannot be deleted.

Once defined, routes and filters can be created for the new logical interface using the other **eth** commands in this section. To list the currently defined logical interfaces, use the **eth list** command (page 181). To remove a logical interface, use an **eth delete** command (page 168).

**Note:** This command requires a **save** and **reboot** before it takes effect.

| **eth add** <*port#*>:<*logical#*> |
|---|

*port#* Ethernet interface (0 for a single-port router; 0 or 1 for a dual-port router).

*logical#* New logical interface number. It cannot be 0 because logical interface 0 always exists.

**Example:** `eth add 0:1`

## ETH DELETE

Deletes a logical interface from an Ethernet port. For more information, see Multiple IP Subnets, on page 76

When you delete a logical interface, all information defined for that interface, such as routes and filters, is deleted automatically.

To list all currently defined logical interfaces, use the **eth list** command (page 181).

**Note:** Logical interface 0 cannot be deleted.

**Note:** This command takes effect immediately; however, if the change is not saved before the next **reboot**, the deletion is lost and the deleted interface reappears after the reboot.

| **eth delete** <*port#*>:<*logical#*> |
|---|

*port#* Ethernet interface (0 for a single-port router; 0 or 1 for a dual-port router).

*logical#* Logical interface number. (It cannot be 0.)

**Example:** `eth delete 0:1`

## ETH IP ADDR

Defines the IP address and subnet mask for an Ethernet port or logical interface.

| **eth ip addr** <*ipaddr*> <*ipnetmask*> [<*interface*>] |
|---|

*ipaddr* Ethernet LAN IP address (4 decimals separated by periods.)

*ipnetmask* IP network mask (4 decimals separated by periods.)

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Examples:**

The following command sets the IP address and subnet mask for the default Ethernet interface (0:0).

```
eth ip addr 192.168.1.254 255.255.255.0
```

The following command sets the IP address and subnet mask for logical interface 1 on Ethernet port 0.

```
eth ip addr 10.0.27.1 255.255.255.0 0:1
```

## ETH IP ADDROUTE

Adds a route to the default routing table for the Ethernet interface.

This command is needed only if the system does not support RIP (see IP (RIP) Protocol Controls, on page 80 and the **eth ip options** command, on page 178).

**Note:** This command requires a **save** and **reboot** before it takes effect.

**eth ip addRoute** <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*interface*>]

| | |
|---|---|
| *ipaddr* | Ethernet LAN IP address (4 decimals separated by periods). |
| *ipnetmask* | IP network mask (4 decimals separated by periods). |
| *gateway* | IP address (4 decimals separated by periods). |
| *hops* | Number of routers through which the packet must go to get to its destination. |
| *interface* | Ethernet interface through which the packet is sent out. This parameter may be omitted if the router has only one Ethernet interface.<br><br>If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.<br><br>To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1). |

**Examples:**

The following command adds a route to the default routing table for the default Ethernet interface (0:0).

```
eth ip addRoute 10.1.2.0 255.255.255.0 192.168.1.17 1
```

The following command adds a route to the default routing table for logical interface 1 on Ethernet port 0.

```
eth ip addRoute 10.1.3.0 255.255.255.0 10.0.27.20 1 0:1
```

## ETH IP BINDROUTE

Adds an Ethernet route to the named IP virtual routing table.

Duplicate routes are not allowed within a routing table. However, identical routes may be added to different routing tables. For example, the same route may be added to a virtual routing table and to the default routing table.

To list the routes, use the **iproutes** command, page 139. To remove an Ethernet route from a virtual routing table, use the **eth ip unbindRoute** command, page 179.

**Note:** A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

**eth ip bindRoute** <*ipaddr*> <*ipnetmask*> <*hops*> [<*gateway*>] <*tablename*> [<*interface*>]

*ipaddr* Ethernet LAN IP address (4 decimals separated by periods).

*ipnetmask* IP network mask (4 decimals separated by periods).

*hops* Number of routers through which the packet must go to get to its destination.

*gateway* IP address of the gateway (4 decimals separated by periods).

*tablename* IP virtual routing table to which the route is added.

*interface* Ethernet interface through which the packet is sent out. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Example:**

The following commands add a route for IP address 10.1.2.0/255.255.255.0 to four routing tables: ROSA, MIGUEL, FRANCISCO, and the default routing table. The first two routes are for Ethernet interface 0:1 and use gateway 192.168.252.9; the second two are for the default Ethernet interface (0:0) and, therefore, specify another gateway (192.168.252.7).

```
eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.9 ROSA 0:1

eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.9 MIGUEL 0:1

eth ip bindRoute 10.1.3.0 255.255.255.0 1 192.168.252.7 FRANCISCO

eth ip addRoute 10.1.3.0 255.255.255.0 1 192.168.252.7
```

## ETH IP DEFGATEWAY

Assigns an Ethernet default gateway for packets whose destination address does not have a route defined.

This setting is most useful when IP routing is not enabled, in which case the system acts as an IP host (i.e., an end system, as opposed to an IP router).

**Note:** This command requires a **save** and **reboot** before it takes effect.

**Note:** The following command is recommended instead of the **eth ip defgateway** command. It sends packets for all IP addresses to the specified gateway:

**eth ip addRoute 0.0.0.0 255.255.255.0** <*gateway*> **1**

**eth ip defgateway** <*ipaddr*> [<*interface*>]

*ipaddr* Ethernet LAN IP address (4 decimals separated by periods).

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Example:** `eth ip defgateway 192.168.1.1`

## ETH IP DELROUTE

Removes a route from the default routing table that was added using the **eth ip addroute** command.

The route to be deleted is identified by its IP address and mask and its Ethernet interface. To see the remaining routes, use the **iproutes** command (page 139).

**Note:** This command requires a **save** and **reboot** before it takes effect.

| **eth ip delRoute** <*ipaddr*> <*ipnetmask*> [<*interface*>] |
|---|

*ipaddr* Ethernet LAN IP address (4 decimals separated by periods).

*ipnetmask* IP network mask (4 decimals separated by periods).

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Examples:**

The following command deletes the route for IP address 10.9.2.0/255.255.255.0 for the default Ethernet interface (0:0).

```
eth ip delRoute 10.9.2.0 255.255.255.0
```

The following command deletes the route for IP address 10.1.3.0/255.255.255.0 for the Ethernet interface 0:1.

```
eth ip delRoute 10.1.3.0 255.255.255.0 0:1
```

## ETH IP DIRECTEDBCAST

Enables or disables the forwarding of packets sent to the network prefix-directed broadcast address of an interface.

A network prefix-directed broadcast address is the broadcast address for a particular network. For example, a network's IP address is 192.168.254.0 and its mask is 255.255.255.0. Its network prefix-directed broadcast address is 192.168.254.255.

| eth ip directedbcast on \| off |
|---|

**on** Enables the forwarding of packets.

**off** Disables the forwarding of packets.

**Example:** `eth ip directedbcast on`

## ETH IP DISABLE

Disables IP routing across the Ethernet LAN. This commands acts as a master switch allowing you to disable all IP routing for testing or control purposes.

**Note:** This command requires a **save** and **reboot** before it is effective.

| eth ip disable |
|---|

**Example:** `eth ip disable`

## ETH IP ENABLE

Enables IP routing across the Ethernet LAN. This command acts as a master switch allowing you to re-enable all IP routing.

**Note:** This command requires a **save** and **reboot** before it is effective.

| eth ip enable |
|---|

**Example:** `eth ip enable`

## ETH IP FILTER

Manages the IP filters for the Ethernet interface(s). The filters are used to screen IP packets.

Each Ethernet interface can have its own set of filters. The intended interface is designated at the end of the filter command. If the router has two physical Ethernet interfaces (an Ethernet hub router), the interface is designated by its port number (0 or 1). If logical interfaces have been defined to provoide service to multiple IP subnets, the logical interface number is also specified (*port #:<logical #*, for example, 0:1).

The set of filters for an interface can be of three types: Input, Forward, and Output. For more information on how these three types of filters are applied, refer to *Chapter 4. IP Filtering on page 103*..

**Note:** IP filters take effect immediately upon entry. They can even affect the current connection that you are using to enter commands. Unlike other configuration changes, you do not need to **save** and **reboot**.

| **eth ip filter** *<command> <type> <action> [<parameters>] [<interface>]* |
|---|

The following commands are provided for managing IP filters for an Ethernet interface:

**eth ip filter append** [*<line number>*] *<type> <action>* [*<parameters>*] *[<interface>]*

Appends a filter to the list of filters for this *<type>* and *<interface>*. The filter is specified by the *<action>* and optional *<parameters>*.

If no line number is specified, the filter is appended to the end of the list; otherwise, it is appended after the specified line. To see the line numbers, use the **eth ip filter list** command. Filters are used in the order they appear in their list.

**eth ip filter insert** *<type> <action> <parameters> [<interface>]*

Inserts a filter in the list of filters for this *<type>* and *<interface>*. The filter is specified by the *<action>* and optional *<parameters>*.

If no line number is specified, the filter is inserted at the beginning of the list; otherwise, it is inserted before the specified line. To see the line numbers, use the **eth ip filter list** command. Filters are used in the order they appear in their list.

**eth ip filter delete** *<type> <action> <parameters> [<interface>]*

Deletes the first filter that matches the filter specified on the command.

**eth ip filter flush** [*<first line> [<last line>]*] *<type> [<interface>]*

Deletes a range of filters from the list for this *<type>* and *<interface>*.

If no line numbers are specified, all filters in the list are deleted. If only the first line number is specified, all filters from that line to the end are deleted. To see the line numbers, use the **eth ip filter list** command. Filters are used in the order they appear in their list.

**eth ip filter clear** [*<first line> [<last line>]*] [*<type>*] *<clear arg> [<interface>]*

Resets the counters for the specified filters. A filter has a counter if the **-c** parameter was specified when the filter was defined.

You can specify the filters whose counters are to be reset by their line number range and type (input, output, or forward). If no type is specified, the counters for all filters for the interface are reset. If no line numbers are specified, the counters for all filters for that type and interface are reset. If only the first line number is specified, all counters for filters from that line to the end of the list are reset. To see the line numbers and counters, use the **eth ip filter list** command.

**eth ip filter check** *<type> <parameters> [<interface>]*

Checks the action that would be taken if a packet with the specified parameters was compared with the list of filters defined for the specified type and interface. For example, the command

**eth ip filter check input -p TCP 1**

would check what action (accept, drop, reject, inipsec, outipsec) would be taken for a TCP packet after it was compared with the list of input filters defined for port 1.

**eth ip filter list** *<type> [<interface>]*

Lists all filters of the specified *<type>* defined for the specified *<interface>*.

**eth ip filter watch** <on | off> [-q | -v] *[<interface>]*

Turns on or turns off the console watch for the interface. If the watch is on, a message is printed to the console serial port when a packet is dropped or rejected.

However, if the parameter **-q** (quiet) was specified for a filter, no message is printed when that filter matches a packet. If the parameter **-v** (verbose) was specified for a filter, a message is printed whenever that filter matches a packet, regardless of the filter action.

To see the messages, Telnet to the router and enter **system log start**. The watch does not continue after a reboot; to resume the watch after a reboot, you must enter the **eth ip filter watch on** command again.

The filter *type* specifies at which point the filter is compared to the IP packet (see the illustration under IP Filtering, on page 103):

**input** Filter is used when the packet enters the interface, before any IP address translation is performed.

**forward** Filter is used, after any IP address translation, but before routing is performed.

**output** Filter is used after routing and IP address translation have been performed, just before the packet is sent out an interface.

If the packet matches the filter, the specified *action* is performed:

**accept** The packet is allowed to proceed for further processing.

**drop** The packet is discarded, without sending an ICMP (Internet Control Management Protocol) error message.

**reject** The packet is discarded and an ICMP error message is returned to the sender.

**inipsec** The packet is passed to IPSec for decrypting. The filter is intended to match packets coming from the other IPSec gateway. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 119).

**outipsec** The packet is passed to IPSec so it can be encrypted and sent to the other IPSec gateway. The filter is intended to match packets coming from the local protected network. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 119).

The following *parameters* specify the characteristics that an IP packet must have in order to match the filter. A filter can require any or all of these characteristics.

**-p <***protocol***> | TCP | UDP | ICMP**
The packet must have the specified protocol. If no protocol is specified, the filter matches *every* protocol.

**-sa <***first source ip addr***>**[:**<***last source ip addr***>**]
The packet must have a source IP address within the specified address range. If only one address is specified, the packet must have that source IP address. If no source IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

**-sm <***source ip mask***>**
The filter uses the specified mask when comparing the <*first source ip addr*>...<*last source ip addr*> with the source IP address in the IP packet. If no source mask is specified, the mask used is 255.255.255.255.

**-sp <***ICMP type***>** | **<***first source port***>**[**:<***last source port***>**]
The packet must have a source port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that source port. If no source port is specified, the filter matches any source port in the range 0**:**0xffff.

-**da <***first dest ip addr***>**[:**<***last dest ip addr***>**]
The packet must have a destination IP address within the specified address range. If only one address is

specified, the packet must have that destination IP address. If no destination IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

**-dm <***dest ip mask***>**
The filter uses the specified mask when comparing the <*first dest ip addr*>...<*last dest ip addr*> with the destination IP address in the IP packet. If no destination mask is specified, the mask used is 255.255.255.255.

**-dp <***ICMP type***>** | **<***first dest port***>**[**:<***last dest port***>**]
The packet must have a destination port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that destination port. If no destination port is specified, the filter matches any destination port in the range 0**:**0xffff.

**-tcp syn|ack|noflag**
If the IP packet is a TCP packet, the filter matches the packet only if the packet flag settings are as specified. If no **-tcp** option is specified for the filter, flag settings are not checked.

Specify **syn** if the TCP SYN flag must be set. Specify **ack** if the TCP ACK flag must be set.Specify **noflag** if neither the SYN flag nor the ACK flag can be set.

**Note:** You may specify *more* than one **-tcp** option for the IP filter.

For example, for the IP filter to match the initiation of a TCP connection, specify **-tcp syn**. The filter will match TCP packets that have the TCP SYN flag set but *not* the TCP ACK flag set. For the filter to match the response to initiation of a TCP connection, specify **-tcp syn** *and* **-tcp ack**. The filter will match only TCP packets with *both* the TCP SYN and TCP ACK flags set.

The following *parameters* request additional filter options.

**-b**
This option requests that this filter be compared *twice* with each packet. The first time the source filter information is matched against the source information in the IP packet and the destination filter information is matched against the destination information in the IP packet. The second time the source filter information is matched against the destination information in the IP packet and the destination filter information is matched against the source information in the IP packet.

**-c <***count of times rule used***>**
This option requests a counter for this filter. If specified, a count is kept of how many IP packets have matched this filter since the router was rebooted. To see the current count for a filter, use the **eth ip filter list** command. To clear a counter, use the **eth ip filter clear** command.

**-ipsec** <*IPSec record name*>

Use this option when the *action* specified is **inipsec** or **outipsec**. It specifies the IPSec Security Association that uses the filter.

**-q** or **-v**

Specify one of these options to determine when watch messages are printed for this filter.

If neither **-q** or **-v** are specified for the filter, and an **eth ip filter watch on** command is entered for the interface, a message is printed to the console serial port each time this filter causes a packet to be dropped or rejected.

If **-q** (quiet) is specified, no messages are printed for this filter, even if the filter causes a packet to be dropped or rejected.

If **-v** (verbose) is specified, a message is printed every time this filter matches a packet, regardless of the filter action.

The optional *interface* determines which Ethernet interface the filter applies to.

If the router has only one Ethernet interface, <*interface*> may be omitted.

If the router has two physical Ethernet interfaces (that is, a dual-port router), you must specify the port by its number (0 or 1).

If logical interfaces have been defined for the physical Ethernet interface, the port number and the logical interface number are specified (<*port #*>:<*logical #*>, for example, 0:1).

**Examples:**

```
eth ip filter flush input 0
```
This command deletes all IP filters of type “Input” on the Ethernet interface 0.

```
eth ip filter append forward deny
```
This command denies the forwarding of all IP traffic. This IP filter is useful as the "last" IP filter in a default action.

## ETH IP FIREWALL

The router supports IP Internet Firewall Filtering to prevent unauthorized access to your system and network resources from the Internet. This filter discards packets received from the WAN that have a source IP address recognized as a local LAN address. This command sets Ethernet Firewall Filtering on or off and allows you to list the active state.

**Note 1:** This command requires a **save** and **reboot** before it takes effect.

**Note 2:** To perform Firewall Filtering, IP routing must be enabled.

| **eth ip firewall** on \| off \| list |
|---|

| | |
|---|---|
| on | Sets firewall filtering on. IP routing *must* also be enabled for filtering to be performed. |
| off | Sets firewall filtering off. |
| list | Lists the current status of firewall filtering. |

**Example:**
```
# eth ip firewall list
The Internet firewall filter is currently on.
0 offending packets were filtered out.
```

## ETH IP MGMT

This command assigns to an Ethernet interface an IP address which is to be used for management purposes only and not for IP address translation. This management IP address is generally a private network address used solely by the ISP.

The management IP address is separate from the IP address used for IP address translation. The IP address used for address translation is generally a public IP address valid on the Internet. It is set by the **eth ip addr** command (page 168).

**Note:** The management address is not effective until after the next **save** and **reboot**.

**Note:** To use the management address as the source address for a ping, you must specify it using the **-I** option on the **ping** command (page 141). For example, to use management address 192.168.1.2 when pinging destination address 192.168.100.100, specify:

```
ping -I 192.168.1.2 192.168.100.100
```

**Note:** To use the management address as the source address for a copy, you must specify both the source and destination addresses on the **copy** command (page 146).

To list the current management address for the Ethernet interface, if any, use the **eth list** command (page 181). To set a management address for the WAN interface, see **remote setMgmtIpAddr** (page 203).

**eth ip mgmt** *<ipaddr> <ipnetmask> [<interface>]*

*ipaddr* IP address (4 decimals separated by periods).

*ipnetmask* IP subnet mask (4 decimals separated by periods).

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (*<port #>:<logical #>*, for example, 0:1).

**Example:**

```
# eth ip mgmt 10.0.0.2 255.255.255.0 0:1
# save
# reboot
```

## ETH IP MTU

Sets the maximum transfer unit for the Ethernet interface. The default is 1500 bytes.

You can set the MTU size to less than 1500 bytes, but you cannot set the MTU to greater than 1500 bytes, even if you specify a larger value on an **eth ip mtu** command. (RFC 1042 recommends 1500 bytes as the maximum MTU for an Ethernet network.)

To see the current MTU size for an interface that has IP enabled, use the **ipifs** command (page 139).

**eth ip mtu** *<size>* [*<interface>*]

*size* Maximum number of bytes that can be transferred as a unit.

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Example:**

The following command decreases the MTU size for Ethernet interface 0:1 to 1400 bytes.

```
eth ip mtu 1400 0:1
```

## ETH IP OPTIONS

RIP is a protocol used for exchanging IP routing information among routers. The following RIP options allow you to set IP routing information protocol controls on the local Ethernet LAN. (See IP (RIP) Protocol Controls, on page 80)

**Note:** This command requires a **save** and **reboot** before it takes effect.

| **eth ip options** <*option*> on \| off [<*interface*>] |
|---|

*option* Must be one of the following:

**rxrip** Receive and process IP RIP-1 compatible and RIP-2 broadcast packets from the Ethernet LAN. Also receive and process RIP-2 packets that are multicast as defined by the **eth ip ripmulticast** command. Set this option if the local router is to discover route information from the Ethernet LAN. The default is on.

**rxrip1** Receive and process RIP-1 packets only.

**rxrip2** Receive and process RIP-2 packets only.

**rxdef** Receive the default route address from the Ethernet LAN. The default is on. This option is useful if you do not want to configure your router with a default route.

**txrip** Transmit RIP-1 compatible broadcast packets and RIP-2 multicast packets over the Ethernet LAN. The default is on.

**txrip1** Transmit broadcast RIP-1 packets only.

**txrip2** Transmit multicast RIP-2 packets only.

**txdef/avdfr** Advertise this router as the default router over the Ethernet LAN (provided it has a default route). The default is on. Set this to off if another router on the local LAN is the default router.

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Example:** `eth ip options avdfr off`

## ETH IP RIPMULTICAST

Changes the multicast address for RIP-1 compatible and RIP-2 packets. The default address is 224.0.0.9.

For more information, see IP (RIP) Protocol Controls, on page 80.

| **eth ip ripmulticast** <*ipaddr*> |
|---|

*ipaddr* IP address of the remote network or station (4 decimals separated by periods).

**Example:** `eth ip ripmulticast 239.192.0.9`

## ETH IP UNBINDROUTE

Removes an Ethernet route from the named IP virtual routing table.

To list the routes, use the **iproutes** command, page 139. To add an Ethernet route to a virtual routing table, use the **eth ip bindRoute** command.

**Note:** A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

| **eth ip unbindRoute** <*ipaddr*> <*tablename*> [<*interface*>] |
|---|

*ipaddr* Ethernet LAN IP address (4 decimals separated by periods).

*tablename* IP virtual routing table from which the route is removed.

*interface* Ethernet interface. This parameter may be omitted if the router has only one Ethernet interface.

If the router has two physical Ethernet interfaces (an Ethernet hub router), the port number (0 or 1) must be specified.

To specify a logical interface other than logical interface 0, specify both the port number and the logical interface number (<*port #*>:<*logical #*>, for example, 0:1).

**Example:**

The following commands remove Ethernet routes from virtual routing table ROSA. The first deleted route is for IP address 10.1.2.0 and the default Ethernet interface (0:0). The second deleted route is for IP address 10.1.3.0 and the logical Ethernet interface 0:1.

```
eth ip unbindRoute 10.1.2.0 ROSA

eth ip unbindRoute 10.1.3.0 ROSA 0:1
```

## ETH IPX ADDR

Sets the IPX network number for the Ethernet LAN connection.

| **eth ipx addr** <*ipxnet*> [*port#*] |
|---|

*ipxnet* IPX network number represented by 8 hexadecimal characters.

*port#* Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

**Example:** `eth ipx addr 123`

## ETH IPX DISABLE

Disables IPX routing across the Ethernet LAN. This acts as a master switch allowing you to disable IPX Routing for testing or control purposes.

**Note:** This command requires a reboot.

| **eth ipx disable** [*port#*] |
|---|

*port#* Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

**Example:** `eth ipx disable`

## ETH IPX ENABLE

Enables IPX routing across the Ethernet LAN. This acts as a master switch that allows you to enable IPX routing.

**Note:** This command requires a reboot.

| **eth ipx enable** [*port#*] |
|---|

*port#* Port number of the Ethernet LAN. This number must be 0 or 1, or it may be omitted.

**Example:** `eth ipx enable`

## ETH IPX FRAME

Sets the frame encapsulation method. The default is 802.2.

| **eth ipx frame** <*type*> |
|---|

*type* 802.2 (DEC standard)
802.3 (Intel standard)
dix (Xerox/Ethernet II standard)

**Example:** `eth ipx frame 802.3`

## ETH LIST

Lists information about the Ethernet interfaces including the status of bridging and routing, IP protocol controls, and IP address and subnet mask.

| **eth list** [<*interface*>] |
|---|

*interface* Ethernet interface for which information is listed. If the parameter is omitted, information is listed for all Ethernet interfaces in the router.

For a dual-port router, you may specify the port number (0 or 1).

If logical interfaces are defined, you may specify a port and logical interface number (<*port #*>:<*logical interface #*>, such as 0:1).

**Example:**

```
# eth list
GLOBAL BRIDGING/ROUTING SETTINGS:
  Bridging enabled..................... no
    Exchange spanning tree with dest... yes
  IP Routing enabled................... yes
    Multicast forwarding enabled....... no
    Firewall filter enabled............ yes
    Directed Broadcasts Allowed........ no
    RIP Multicast address.............. default
  IPX Routing enabled.................. no

ETHERNET INFORMATION FOR <ETHERNET/0>
  Hardware MAC address................. 00:20:6F:02:98:04
  Send IP RIP to the LAN.................. no
    Advertise me as default router..... yes
  Process IP RIP packets received...... no
    Receive default route by RIP....... yes
  IP filters defined................... no
  IP address/subnet mask............... 192.168.0.101/255.255.255.0
  Static Ethernet routes defined....... 1
    IP address/subnet mask.............. 0.0.0.0/0.0.0.0
    IP gateway/metric.................. 192.168.0.252/1
  Virtual Ethernet routes defined...... none
  IPX External network number.......... 00000000
  IPX Frame type....................... 802.2
  MTU.................................. default
```

# Remote Router Access Configuration (REMOTE)

The following commands allow you to add, delete, and modify remote routers to which the target router can connect. Remote router information that can be configured includes:

- PVC numbers
- Security authentication protocols and passwords
- WAN IP/ IPX addresses
- IP routes
- IPX routes and SAPS
- Remote bridging addresses and bridging control
- Host mapping
- Encryption (optional)
- IP filtering (optional)
- L2TP tunneling (optional)

## REMOTE ?

Lists the supported keywords. (The list varies depending on the router model.)

| **remote ?** |
|---|

**Example:**

```
# remote ?
Sub-commands for remote:
?                   help                  add
del                 delete                list
enable              disable               setAuthen
enaAuthen           disAuthen             setPasswd
setOurPasswd        delOurPasswd          setOurSysName
delOurSysName       listPhones            setLNS
setL2TPClient       setProtocol           setPVC
setATMnsap          delATMnsap            setATMTraffic
setPhone            delPhone              setBWThresh
setBod              addCaller             delCaller
setMaxLine          setMinLine            setTimer
addHostMapping      delHostMapping        addServer
delServer           setIPTranslate        setIPslavePPP
setPPPOptions       ipfilter              blockNetBIOS
setCompression      stats                 statsclear
setRmtIpAddr        addIproute            delIproute
bindIPVirtualRoute  unbindIPVirtualRoute  setIpOptions
listIproutes        setIpxaddr            addIpxroute
delIpxroute         listIpxroutes         addIpxsap
```

```
delIpxsap          listIpxsaps          setIpxOptions
listBridge         setBrOptions         addBridge
delBridge          enaBridge            disBridge
setmtu
```

## REMOTE ADD

Adds a remote router entry into the remote router database.

| **remote add** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string). The name is case-sensitive.

**Example:** `remote add HQ`

## REMOTE ADDHOSTMAPPING

Remaps a range of local LAN IP addresses to a range of public IP addresses on a *per-remote-router basis*. These local addresses are mapped one-to-one to the public addresses.

**Note:** The range of public IP addresses is defined by *<first public addr>* only. The rest of the range is computed automatically (from *<first public addr>* to *<first public addr>* + number of addresses remapped - 1) inclusive.

| **remote addHostMapping** <*first private addr*><*second private addr*><*first public addr*><*remoteName*> |
|---|

*first private addr* First IP address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods.

*second private addr* Last address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods.

*first public addr* Defines the range of public IP addresses, in the format of 4 decimals separated by periods.

The rest of the range is computed automatically.

*remoteName* Name of the remote router (character string).

**Example:** `remote addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 HQ`

## REMOTE ADDIPROUTE

Adds an IP address route to a network or station on the LAN connected beyond the remote router. The route is added to the default routing table.

The target router's routing table must be seeded statically to access networks and stations beyond this remote router. After the connection is established, standard RIP update packets can dynamically add to the routing table. Setting this address is not required if a target router never connects to the remote router *and* the remote router supports RIP.

**Note:** Changes to the default routing table require a **save** and **reboot** before they take effect.

**remote addIpRoute** <*ipaddr*> <*ipnetmask*> <*hops*> [<*ipgateway*>] <*remoteName*>

*ipaddr* IP address of the remote network or station (4 decimals separated by periods).

*ipnetmask* IP network mask of the remote network or station (4 decimals separated by periods).

*hops* Perceived cost to reach the remote network or station by this route (number between 1 and 15).

*ipgateway* Address of a router on the remote LAN (4 decimals separated by periods).
Enter a gateway only if you are configuring a MER interface. Check with your system administrator for details.

*remoteName* Name of the remote router (character string).

**Examples:**

The first two addresses in the list represent subnetworks, the third is a class B network, the fourth is a host, and the fifth address is the default route. The fifth command adds the default route when the WAN interface is a point-to-point interface; the sixth command adds the default route when the WAN interface is a broadcast interface.

```
remote addIpRoute 10.1.210.64 255.255.255.192 1 HQ

remote addIpRoute 10.1.210.032 255.255.255.224 1 HQ

remote addIpRoute 172.17.0.0 255.255.0.0 2 HQ

remote addIpRoute 10.1.210.072 255.255.255.255 1 HQ

remote addIpRoute 0.0.0.0 255.255.255.255 1 HQ

remote addIproute 0.0.0.0 255.255.255.255 1 172.16.10.1 HQ
```

## REMOTE ADDIPXROUTE

Adds an IPX route for a network or station on the LAN network connected beyond the remote router. The target router's routing information table must be seeded statically to access networks and stations beyond this remote router. After the connection is established, standard RIP update packets will dynamically add to the routing table. (Setting this address is not required if a target router never connects to the remote router and the remote router supports RIP.)

**Note:** A **reboot** command must be performed on the target router for the addition of a static route to take effect.

**remote addIpxRoute** <*ipxNe#*> <*metric*> <*ticks*> <*remoteName*>

*ipxNe#* IPX network number represented by 8 hexadecimal characters.

*metric* Number of routers through which the packet must go to get to the network/station.

*ticks* Number in 1/8 seconds which is the estimated time delay in reaching the remote network or station.

*remoteName* Name of the remote router (character string).

**Example:** `remote addIpxRoute 456 1 4 HQ`

## REMOTE ADDIPXSAP

Adds an IPX SAP to the server information table for a service on the LAN network connected beyond the remote router. The target router's SAP table must be seeded statically to access services beyond this remote router. After the connection is established, standard SAP broadcast packets will dynamically add to the table.

**Note:** A **reboot** must be performed on the target router for the addition of a SAP to take effect.

**remote addIpxSap** <*servicename*> <*ipxNet* > <*ipxNode*> <*socket*> <*type*> <*hops*> <*remoteName*>

| | |
|---|---|
| *servicename* | Name of server. |
| *ipxNet* | IPX network number represented by 8 hexadecimal characters. |
| *ipxNode* | IPX node address represented by 12 hexadecimal characters. |
| *socket* | Socket address of the destination process within the destination node. The processes include services such as file and print servers. |
| *type* | Number representing the type of server. |
| *hops* | Number of routers through which the packet must go to get to the network/station. |
| *remoteName* | Name of the remote router (character string). |

**Example:** `remote addIpxSap Fileserver 010a020b 0108030a0b0c 451 HQ`

## REMOTE ADDSERVER

This Network Address Translation (NAT) command is used to add a server's IP address (on the LAN) associated with this remote router for a particular protocol.

**remote addServer <***ipaddr*>|discard|me <*protocolid***>** |tcp|udp<*first port*> |ftp|telnet|smtp|snmp|http [<*last port*> [<*first private port*>]] <*remoteName*>

| | |
|---|---|
| *ipaddr* | IP address of the host selected as server in the format of 4 decimals separated by periods |
| discard | Discards the incoming server request. |
| me | Sends the incoming server request to the local router, regardless of its IP address. |
| *protocolid* | Protocol used by the selected server; can be **tcp** or **udp.** |
| *first port* | First or only port as seen by the remote end**.** Port used by the selected server;<br>Can be a string such as **ftp, telnet, smtp, snmp**, or **http**, or a numeric value between 0 and 65,535. A numeric value of 0 will match any port. |
| *last port* | If specified, this parameter is used with <*first port*> to specify a range of ports as seen by the remote end for the server on the LAN. |
| *first private port* | If specified, this is a port remapping of the incoming request from the remote end. |

*remoteName* Name of the remote router (character string).

**Example:**
```
remote addServer 192.168.1.5 tcp smtp
remote addServer 192.168.1.10 tcp 9000 9000 telnet router2
```

## REMOTE BINDIPVIRTUALROUTE

Adds a remote route to the named IP virtual routing table.

To list the remote routes, use the **remote listIProutes** command, page 196. To remove a route from a virtual routing table, use the **remote unbindIPVirtualRoute** command, page 207.

**Note:** A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

**remote bindIPVirtualRoute** <*ipaddr*> <*ipnetmask*> <*hops*> *[*<*ipgateway*>*]* <*tableName*> <*remoteName*>

*ipaddr* IP address of the remote network or station (4 decimals separated by periods).

*ipnetmask* IP network mask of the remote network or station (4 decimals separated by periods).

*hops* Perceived cost in reaching the remote network or station by this route (number between 1 and 15).

*ipgateway* Address of a router on the remote LAN (4 decimals separated by periods).
Enter a gateway only if you are configuring a MER interface.

*tableName* IP virtual routing table to which the route is added.

*remoteName* Name of the remote router (character string).

**Example:**

The following command adds a route to virtual routing table FRANCISCO. The route is to IP address 10.1.2.0/255.255.255.0 and goes through remote router HQ.

```
remote bindIPVirtualRoute 10.1.2.0 255.255.255.0 1 francisco HQ
```

## REMOTE BLOCKNETBIOS

This command will filter out all NetBIOS packets over this WAN connection.

**remote blockNetBIOS** onloff <*remoteName*>

## REMOTE DEL

Deletes a remote router entry from the remote router database.

**remote del** <*remoteName*>

*remoteName* Name of the remote router (character string).

**Example:** `remote del HQ`

## REMOTE DELATMNASP

This command deletes the ATM snap setting.

**remote delATMNasp** <*remoteName*>

## REMOTE DELENCRYPTION

Deletes encryption files associated with a remote router.

**remote delEncryption** <*remoteName*>

*remoteName* Name of the remote router (character string).

**Example:** `remote delEncryption HQ`

## REMOTE DELHOSTMAPPING

Undoes an IP address/host translation (remapping) range that was previously established with the command **remote addhostmapping** on a *per-remote-router basis*.

**remote delHostMapping** <*first private addr*> <*second private addr*> <*first public addr*> <*remoteName*>

*first private addr* First IP address in the range of IP address, in the format of 4 decimals separated by periods.

*second private addr* Last address in the range of IP address, in the format of 4 decimals separated by periods.

*first public addr* Defines the range of public IP addresses, in the format of 4 decimals separated by periods. The rest of the range is computed automatically.

*remoteName* Name of the remote router (character string).

**Example:** `remote delHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 HQ`

## REMOTE DELIPROUTE

Deletes an IP address route for a network or station on the LAN connected beyond the remote router. The route is deleted from the default routing table.

**Note:** Changes to the default routing table require a **save** and **reboot** before they take effect.

**remote delIpRoute** <*ipaddr*> <*remoteName*>

*ipaddr* IP address of the remote network or station (4 decimals separated by periods).

*remoteName* Name of the remote router (character string).

**Example:** `remote delIpRoute 10.1.2.0 HQ`

## REMOTE DELIPXROUTE

Deletes an IPX address for a network on the LAN connected beyond the remote router.

**Note:** The **reboot** command must be issued on the target router for a deleted static route to take effect.

**remote delIpxroute** <*ipxNet*> <*remoteName*>

*ipxNet* IPX network number represented by 8 hexadecimal characters.

*remoteName* Name of the remote router (character string).

**Example:** `remote delIpxRoute 010a020b HQ`

## REMOTE DELIPXSAP

Deletes an IPX service on the LAN network connected beyond the remote router.

**Note:** The **reboot** command must be issued on the target router for a deleted service to take effect.

**remote delIpxSap** <*servicename*> <*remoteName*>

*servicename* Name of server

*remoteName* Name of the remote router (character string).

**Example:** `remote delIpxSap Fileserver HQ`

## REMOTE DELOURPASSWD

Removes the unique CHAP or PAP authentication password entries established by the command **remote setOurPasswd.**

| **remote delOurPasswd** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote delOurPasswd HQ`

## REMOTE DELOURSYSNAME

Removes the unique CHAP or PAP authentication system name entries established by the command **remote setOurSysName.**

| **remote delOurSysName** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote delOurSysName HQ`

## REMOTE DELSERVER

This Network Address Translation (NAT) command deletes an entry created by the **remote addServer** command. Refer to the section Server Configuration, on page 92, for detailed information.

| **remote delServer <***ipaddr***>**\|discard\|me <*protocolid***>** \|tcp\|udp <*first port*> \|ftp\|telnet\|smtp\|snmp\|http [<*last port*> [<*first private port*>]] <*remoteName*> |
|---|

*ipaddr* IP address of the host selected as server in the format of 4 decimals separated by periods.

discard Discards the incoming server request.

me Sends the incoming server request to the local router, regardless of its IP address.

*protocolid* Protocol used by the selected server; can be **tcp** or **udp.**

*first port* First or only port as seen by the remote end**.** Port used by the selected server. Can be as string such as **ftp, telnet, smtp, snmp**, or **http**, or a numeric value between 0 and 65,535. A numeric value of 0 will match any port.

*last port* If specified, this is used with <*first port*> to specify a range of ports as seen by the remote end for the server on the LAN.

*first private port* If specified, this is a port remapping of the incoming request from the remote end.

*remoteName* Name of the remote router (character string).

**Example:** `remote delServer 192.168.1.5 tcp ftp router1`

## REMOTE DISABLE

Disables communications with the remote router. This command allows you to enter routers into the remote router database, but it sets them inactive.

**Note:** The routing information defined for <*routerName*> is still in effect when the entry is disabled until you save and reboot. However, no calls will be made to that remote router.

| **remote disable** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote disable HQ`

## REMOTE DISAUTHEN

This command is intended for situations where third-party routers cannot be authenticate;: the target router will not attempt to authenticate the remote router.

| **remote disAuthen** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote disAuthen HQ`

## REMOTE DISBRIDGE

Disables bridging from the target router to the remote router.

**Note:** This command requires rebooting the target system for the change to take effect.

| **remote disBridge** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote disBridge HQ`

## REMOTE ENAAUTHEN

With this command the target router will try to negotiate authentication as defined in the remote router's database.

| **remote enaAuthen** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote enaAuthen HQ`

## REMOTE ENABLE

Enables communications with the remote router. This command allows you to activate the entry in the remote router database when you are ready.

| **remote enable** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote enable HQ`

## REMOTE ENABRIDGE

Enables bridging from the target router to the remote router. This command requires rebooting the target system for the change to take effect.

| **remote enaBridge** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote enaBridge HQ`

## REMOTE IPFILTER

This command manages the IP filters on the WAN interface. The filters screen IP packets at the interface level.

You can define filters for any entry in the remote router database. The remote entry is referenced by the name specified on the **remote add** command that created the entry.

The set of filters for a remote router entry can be of three types: Input, Forward, and Output. For more information on how these three types of filters are applied, refer to *Chapter 4. IP Filtering on page 103*.

**Note:** IP filters take effect immediately upon entry. They can even affect the current connection that you are using to enter commands. Unlike other configuration changes, you do not need to **save** and **reboot**.

| **remote ipfilter** <*command*> <*type*> <*action*> <*parameters*> <*remoteName*> |
|---|

The following commands are provided for managing IP filters for the WAN interface:

**remote ipfilter append** [<*line number*>] <*type*> <*action*> [<*parameters*>] <*remoteName*>

Appends a filter to the list of filters for this <*type*> (Input, Output, or Forward) for this remote router entry.

If no line number is specified, the filter is appended to the end of the list; otherwise, it is appended after the specified line. To see the line numbers, use the **remote ipfilter list** command. Filters are used in the order they appear in their list.

**remote ipfilter insert** *<type> <action> <parameters> <remoteName>*

Inserts a filter in the list of filters for this *<type>* (Input, Output, or Forward) for this remote router entry.

If no line number is specified, the filter is inserted at the beginning of the list; otherwise, it is inserted before the specified line. To see the line numbers, use the **remote ipfilter list** command. Filters are used in the order they appear in their list.

**remote ipfilter delete** *<type> <action> <parameters> <remoteName>*

Deletes the first filter that matches the filter specified on the command.

**remote ipfilter flush** [*<first line> [<last line>]*] *<type> <remoteName>*

Deletes a range of filters of this *<type>* (Input, Output, or Forward) for this remote router entry.

If no line numbers are specified, all filters in the list are deleted. If only the first line number is specified, all filters from that line to the end are deleted. To see the line numbers, use the **remote ipfilter list** command. Filters are used in the order they appear in their list.

**remote ipfilter clear** [*<first line> [<last line>]*] [*<type>*] *<clear arg> <remoteName>*

Resets the counters for the specified filters. A filter has a counter if the **-c** parameter was specified for the filter.

You can specify the filters whose counters are to be reset by their line number range and type (input, output, or forward). If no type is specified, the counters for all filters for the interface are reset. If no line numbers are specified, the counters for all filters for that type and interface are reset. If only the first line number is specified, all counters for filters from that line to the end are reset. To see the line numbers and counters, use the **remote ipfilter list** command.

**remote ipfilter check** *<type> <parameters> <remoteName>*

Checks the action that would be taken if a packet with the specified parameters was compared with the list of filters defined for the specified type and remote router entry.

For example, the command

```
remote ipfilter check input -p TCP branch1
```

would check what action (accept, drop, reject, inipsec, outipsec) would be taken for a TCP packet after it was compared with the list of input filters defined for remote router branch1.

**remote ipfilter list** *<type> <remoteName>*

Lists all filters of the specified *<type>* (Input, Output, or Forward) for this remote router entry.

**remote ipfilter watch** <on | off> [-q | -v] *<remoteName>*

Turns on or turns off the console watch for this remote router entry. If the watch is on, a message is printed to the console serial port when a packet is dropped or rejected.

However, if the parameter **-q** (quiet) was specified for a filter, no message is printed when that filter matches a packet. If the parameter **-v** (verbose) was specified for a filter, a message is printed whenever that filter matches a packet, regardless of the filter action.

To see the messages, Telnet to the router and enter **system log start**. The watch does not continue after a reboot; to resume the watch after a reboot, you must enter the **remote ipfilter watch on** command again.

The filter *type* specifies at which point the filter is compared to the IP packet (see the illustration under IP Filtering, on page 103):

**input** Filter is used when the packet enters the interface, before any IP address translation is performed.

**forward** Filter is used, after any IP address translation, but before routing is performed.

**output** Filter is used after routing and IP address translation have been performed, just before the packet is sent out an interface.

If the packet matches the filter, the specified *action* is performed:

**accept** The packet is allowed to proceed for further processing.

**drop** The packet is discarded, without sending an ICMP (Internet Control Management Protocol) error message.

**reject** The packet is discarded and an ICMP error message is returned to the sender.

**inipsec** The packet is passed to IPSec for decrypting. The filter is intended to match packets coming from the other IPSec gateway. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE, rather than your own filters, to manage your IP Security (see IPSec (Internet Protocol Security), on page 119).

**outipsec** The packet is passed to IPSec so it can be encrypted and sent to the other IPSec gateway. The filter is intended to match packets coming from the local protected network. Although filters are the mechanism by which packets are passed to IPSec, it is recommended that you use IKE to manage your IP Security (see IPSec (Internet Protocol Security), on page 119).

The following *parameters* specify the characteristics that an IP packet must have in order to match the filter. A filter can require any or all of these characteristics.

**-p <***protocol***> | TCP | UDP | ICMP**
The packet must have the specified protocol. If no protocol is specified, the filter matches *every* protocol.

**-sa <***first source ip addr***>**[:**<***last source ip addr***>**]
The packet must have a source IP address within the specified address range. If only one address is specified, the packet must have that source IP address. If no source IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

**-sm <***source ip mask***>**
The filter uses the specified mask when comparing the <*first source ip addr*>...<*last source ip addr*> with the source IP address in the IP packet. If no source mask is specified, the mask used is 255.255.255.255.

**-sp <***ICMP type***>** | **<***first source port***>**[**:<***last source port***>**]
The packet must have a source port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that source port. If no source port is specified, the filter matches any source port in the range 0**:**0xffff.

-**da <***first dest ip addr***>**[:**<***last dest ip addr***>**]
The packet must have a destination IP address within the specified address range. If only one address is specified, the packet must have that destination IP address. If no destination IP address is specified, the filter matches any address in the range 0.0.0.0**:**255.255.255.255.

**-dm <***dest ip mask***>**
The filter uses the specified mask when comparing the *<first dest ip addr>...<last dest ip addr>* with the destination IP address in the IP packet. If no destination mask is specified, the mask used is 255.255.255.255.

**-dp <***ICMP type***>** | **<***first dest port***>**[**:<***last dest port***>**]
The packet must have a destination port that matches the specified ICMP type or that is within the specified port range. If only one port is specified, the packet must have that destination port. If no destination port is specified, the filter matches any destination port in the range 0**:**0xffff.

**-tcp syn|ack|noflag**
If the IP packet is a TCP packet, the filter matches the packet only if the packet flag settings are as specified. If no **-tcp** option is specified for the filter, flag settings are not checked.

Specify **syn** if the TCP SYN flag must be set. Specify **ack** if the TCP ACK flag must be set.Specify **noflag** if neither the SYN flag nor the ACK flag are to be set.

**Note:** You may specify *more* than one **-tcp** option for the IP filter.

For example, for the IP filter to match the initiation of a TCP connection, specify **-tcp syn**. The filter will match TCP packets that have the TCP SYN flag set but *not* the TCP ACK flag set. For the filter to match the response to initiation of a TCP connection, specify **-tcp syn** *and* **-tcp ack**. The filter will match only TCP packets with *both* the TCP SYN and TCP ACK flags set.

The following *parameters* request additional filter options.

**-b**
This option requests that this filter be compared *twice* with each packet.The first time the source filter information is matched against the source information in the IP packet and the destination filter information is matched against the destination information in the IP packet. The second time the source filter information is matched against the destination information in the IP packet and the destination filter information is matched against the source information in the IP packet.

**-c <***count of times rule used***>**
This option requests a counter for this filter. If specified, a count is kept of how many IP packets have matched this filter since the router was rebooted.To see the current count for a filter, use the **remote ipfilter list** command. To clear a counter, use the **remote ipfilter clear** command.

**-ipsec** *<IPSec record name>*

Use this option when the *action* specified is **inipsec** or **outipsec**. It specifies the IPSec Security Association that uses the filter.

**-q** or **-v**

Specify one of these options to determine when watch messages are printed for this filter.

If neither **-q** or **-v** are specified for the filter, and an **remote ipfilter watch on** command is entered for the interface, a message is printed to the console serial port each time this filter causes a packet to be dropped or rejected.

If **-q** (quiet) is specified, no messages are printed for this filter, even if it causes a packet to be dropped or rejected.

If **-v** (verbose) is specified, a message is printed every time this filter matches a packet, regardless of the filter action.

The *remote name* specifies the entry in the remote router database that the command applies to. The remote name is the name given the entry when it was created by a **remote add** command.

Examples:

This command deletes all IP filters of type Forward for the remote interface **internet**.

```
remote ipfilter flush forward internet
```

Both of the following commands have the same effect: they deny all IP traffic for the remote interface **internet** from the specified destination addresses. The addresses can be specified as 192.168.0.0 masked with 255.255.0.0 or as the range 192.168.0.0 through 192.168.255.255).

```
remote ipfilter append forward drop -da 192.168.0.0 -dm 255.255.0.0 internet
```

```
remote ipfilter append forward drop -da 192.168.0.0:192.168.255.255 internet
```

This command lists all IP filters of type Forward for the remote interface **internet**.

```
remote ipfilter list forward internet
```

## REMOTE LIST

Lists the remote router entry (or all the entries) in the remote router database. The result is a complete display of the current configuration settings for the remote router(s), except for the authentication password/secret.

| **remote list** [<*remoteName*>] |
|---|

*remoteName* Name of the remote router to be listed (character string).If you omit the name, all remote router entries are listed.

**Example:**

```
# rem list internet
INFORMATION FOR <internet>
  Status.............................. enabled
  Our System Name when dialing out.....
  Our Password used when dialing out... no
  Protocol in use..................... PPP
  ATM traffic shaping................. no
  Authentication...................... disabled
  Authentication level required........ PAP
  Use periodic LCP pings.............. yes
  Connection Identifier (VPI*VCI)...... 0*38
  IP address translation.............. off
  IP filters defined.................. no
  Send/Receive Multicast.............. off
  Block NetBIOS Packets............... off
  Compression Negotiation............. off
  IP slave mode (PPP)................. no
  Try to reacquire IP addr (PPP)....... yes
  Source IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Remote IP address/subnet mask........ 0.0.0.0/0.0.0.0
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
```

```
Receive IP RIP from this dest........ no
  Receive IP default route by RIP.... no
Keep this IP destination private..... yes
Total IP remote routes.............. 1
         10.0.0.0/255.255.0.0/1
IPX network number.................. 00000000
Use IPX RIP/SAP (negotiate with PPP): yes
Total IPX remote routes.............. 0
Total IPX SAPs....................... 0
Bridging enabled..................... no
  Exchange spanning tree with dest... no
TX Encryption........................ unknown
RX Encryption........................ unknown
mtu.................................. 1500
```

## REMOTE LISTBRIDGE

Lists the bridging capability from the target router to the remote router.

**remote listBridge** <*remoteName*>

*remoteName* Name of the remote router (character string).

**Example:**

```
# rem listBridge HQ
BRIDGING INFORMATION FOR <HQ>
 Bridging enabled..................... yes
 Exchange spanning tree with dest.... yes
```

## REMOTE LISTIPROUTES

Lists IP information for a remote router or, if the router name is omitted, for all routers in the remote router database. The IP information includes all network or station IP addresses defined for the LAN connected beyond the remote router.

This command lists all routes defined for the remote router, including those defined in the default routing table and in any virtual routing tables.

**remote listIpRoutes** [*remoteName*]

*remoteName* Name of the remote router (character string).

**Example:**

```
remote listIproutes HQ
```

**Example:**

The following command lists routing information for remote router HQ. It lists five routes that use HQ, the first four are in the default routing table and the fifth is in virtual routing table FRANCISCO.

```
# remote listIproutes HQ
IP INFORMATION FOR <HQ>
  Send IP RIP to this dest............. no
    Send IP default route if known..... no
```

```
Receive IP RIP from this dest........ rip-1 compatible
  Receive IP default route by RIP.... no
Keep this IP destination private..... yes
Total IP remote routes.............. 5
         192.168.210.0/255.255.255.0/1
         10.0.0.0/255.0.0.0/1
         172.16.0.0/255.240.0.0/1
         192.168.0.0/255.255.0.0/1
         10.1.2.0/255.255.255.0/1 through <FRANCISCO>
```

## REMOTE LISTIPXROUTES

Lists all network IPX route addresses defined for the LAN connected beyond the remote router. The network number, hop count, and ticks are displayed. If the remote name is not specified, a list of IPX routes is displayed for each remote router in the database.

**remote listIpxroutes** [*remoteName*]

*remoteName* Name of the remote router (character string).

**Example:** `remote listIpxroutes HQ`

**Response:**

```
IPX ROUTE INFORMATION FOR <HQ>
Total IPX remote routes ..............1     00001001/1/4
```

## REMOTE LISTIPXSAPS

Lists all services defined for the LAN connected beyond the remote router. Each service includes the server name, network number, node number, socket number, server type, and hop count. If the remote name is not specified, a list of IPX SAPs is displayed for each remote router in the database.

**remote listIpxsaps** [*remoteName*]

*remoteName* Name of the remote router (character string.)

**Example:** `remote listIpxsaps HQ`

**Response:**

```
IPX SAP INFORMATION FOR <HQ>
Total IPX SAPs ...................  1
SERV312_FP 00001001 00:00:00:00:00:01 0451 0004 1

IPX SAP INFORMATION FOR <ISP>
Total IPX SAPs ...................  0
   SERV312_FP 00001001 00:00:00:00:00:01 0451 0004 1
```

## REMOTE LISTPHONES

Lists the PVC numbers available for connecting to the remote router.

| **remote listPhones** <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote listPhones HQ`

**Response:**

```
PHONE NUMBER(s) FOR <HQ>
Connection Identifier (VPI*VCI)...... 0*38
```

**Note:** If the remote name is not specified, a list of phone numbers is displayed for each remote router in the database.

## REMOTE SETATMNSAP

RFC1577 (Classical IP over ATM) specifies a mechanism to map an ATM Name (called an NSAP) to a PVC. NSAP's are normally not neeed, but if they are used they have a defined syntax defined by either the using ATM or E164 encodings. By convention, octets 2-7 contain a unique identifier for the router such as a MAC address.

In the command remote setATMNSAP the complete 20 octets of the NSAP are specified. If Partial mode has been selected, the router will substitute the MAC address of the router for Octets 2-7. In Full mode no change is made to the NSAP.

| **remote setATMNSAP ATMF\|E164** <*nsap*> <*remoteName*> |
|---|

ATMF specifies ATM forum encoding

E164 specifies ITU E164.

## REMOTE SETATMTRAFFIC

This command applies only to ATM routers. Refer to *Asynchronous Transfer Mode Commands (ATM), on page 210* for more syntax information.

| **remote setATMTraffic scr mbs** <*remoteName*> |
|---|

## REMOTE SETAUTHEN

Sets the authentication protocol used communicate with the remote router. The authentication protocol is the *minimum* security level that the target router must use with the remote router; this level is verified during security negotiation. The router will *always* attempt to negotiate the highest level of security possible (CHAP). The router will not accept a negotiated security level less than this minimum authentication method.

The parameter in the remote router database is used for the local side of the authentication process; this isthe minimum security level used by the target router when it challenges or authenticates the remote router.

**remote setAuthen** <*protocol*> <*remoteName*>

*protocol* **c**hap, pap, or none. The default is pap.

*remoteName* Name of the remote router (character string).

**Example:** remote setAuthen pap HQ

## REMOTE SETBROPTIONS

Sets controls on the bridging process.

**Warning:** Do not change this setting without approval from your system administrator.

**remote setBrOptions** <*option*> on|off <*remoteName*>

*option* stp

Use the spanning tree protocol for bridging. Set this option on only if the bridging peers support the spanning tree protocol and you wish to detect bridging loops. The default is **on**.

**Note:** This adds a 40-second delay each time the ADSL or ATM link comes up; use only if necessary.

*remoteName* Name of the remote router (character string).

**Example:** `remote setBrOptions stp on HQ`

## REMOTE SETCOMPRESSION

Enables or disables compression between the local router and the remote router.

**remote setCompression** on|off <*remoteName*>

on Compression will be negotiated between the local and the remote router *if* both routers are set to perform compression and *if* they both share a common compression protocol.

off Disables compression. The default is off.

*remoteName* Name of the remote router (character string).

**Example:** `remote setCompression on HQ`

## REMOTE SETDLCI

Refer to the section ISDN Digital Subscriber Line (IDSL), on page 220 for further information regarding this command.

## REMOTE SETENCRYPTION (RFC 1969 Encryption)

This command is used to specify a PPP DES (Data Encryption Standard) 56-bit key with fixed transmit and receive keys.

| **remote setEncryption** DESE RX\|TX <*key*> <*remoteName*> |
|---|

| | |
|---|---|
| RX | Receive key |
| TX | Transmit key |
| *key* | Key in the format of an eight-hexadecimal number. |
| *remoteName* | Name of the remote router (character string). |

**Example:**

```
remote setEncryption dese tx 1111111111111111 HQ
remote setEncryption dese rx 2222222222222222 HQ
```

## REMOTE SETENCRYPTION (Diffie-Hellman Encryption)

This command is used to specify encryption based on the Diffie-Hellman key-exchange protocol. Each router possesses an internal encryption file that is associated with a public key providing 768-bit security. The predefined keys can be replaced by the user. The configuration file on the router must have a “num” suffix (e.g., dh96.num).

| **remote setEncryption** DESE_1_KEY\|DESE_2_KEY [<*filename*>] <*remoteName*> |
|---|

| | |
|---|---|
| DESE_1_KEY | Specifies that the same key is used in both directions |
| DESE_2_KEY | Specifies that the keys are different |
| *filename* | Name of the file containing the Diffie-Hellman values. If the file is not specified, default values built into the router’s kernel are automatically selected. |
| *remoteName* | Name of the remote router (character string). |

**Example:**

```
remote setEncryption DESE_1_KEY dh96.num HQ
```

## REMOTE SETIPOPTIONS

RIP is a protocol used for exchanging IP routing information among routers. The following RIP options allow you to set IP routing information protocol controls over a point-to-point WAN.

| **remote setipoptions** <*option*> onloff <*remoteName*> |
|---|

| | |
|---|---|
| *option* | Includes the following choices: |
| **rxrip** | Receive and process IP RIP-1 compatible packets and RIP-2 broadcast packets from the remote site. Also receive and process RIP-2 multicast packets. Set this option if the local router is to discover route information from other sites connected to the remote router. This is useful for hierarchical organizations. If you are connecting to another company or an Internet Service Provider, you may wish to set this option off. The default is off**.** |
| **rxrip1** | Receive and process RIP-1 packets only. |
| **rxrip2** | Receive and process RIP-2 packets only. |
| **rxdef** | Receive default IP route address. Set on, the local router will receive the remote site's default IP route. The default is off**.** |
| **txrip** | Transmit IP RIP-1 compatible broadcast packets and RIP-2 multicast packets to the remote site. Set on, the local router will send routing information packets to the remote site. The default is off. |
| **txrip1** | Transmit broadcast RIP-1 packets only. |
| **txrip2** | Transmit multicast RIP-2 packets only. |
| **txdef** | Transmit the local router's default IP route. Set to on, the local router will send the default route to the remote site. The default is off**.** |
| **private** | Keep IP routes private. Used to prevent advertisement of this route to other sites by the remote router. Used as a security mechanism when the remote site is outside your company (an Internet Service Provider, for example), or whenever you would prefer to keep the identify of the site private. The default is yes**.** |
| **multicast** | Allows the remote router to send and receive IP multicast traffic. |
| *remoteName* | Name of the remote router (character string). |
| **Example:** | `remote setipoptions private on HQ` |

## REMOTE SETIPSLAVEPPP

| **remote setIPSlavePPP** yes\|no <*remoteName*> |
|---|

If SetIPSlaveModePPP is yes, the router will accept the IP address that the remote end informs the router that it has without reguard to how the router was previously configured. If setIPSlaveModePPP is no the router will try to use the address that it was configured for.

Normally there is no need to change the default (no) value of this option, however, in certain situations where the router is managed by another party, (as part of a managed service), it might make sense to set this value to yes to ensure that the central management site always specifies the IP address of the router.

## REMOTE SETIPTRANSLATE

This command is used to control Network Address Translation on a per remote router basis. It allows several PCs to share a single IP address to the Internet. The remote router must assign the source WAN IP address to the routers' local WAN port. This command requires that you define a Source WAN IP Address with the command: **remote setSrcIpAddr**

| **remote setIPTranslate** on\|off <*remoteName*> |
|---|

*remoteName* Name of the remote router (character string).

**Example:** `remote setIPTranslate on HQ`

## REMOTE SETIPXADDR

Sets the IPX network number for the remote WAN connection.

| **remote setIpxaddr** <*ipxNet*> [*port#*] |
|---|

*ixpNet* IPX network number represented by 8 hexadecimal characters.

*port#* Port number of the Ethernet LAN. This number must be either 0 or it may be omitted.

**Example:** `remote setIpxaddr 789 HQ`

## REMOTE SETL2TPCLIENT

This command is specific to L2TP tunnel configuration. Please, refer to the L2TP commands section, L2TP — Virtual Dial-Up Configuration (L2TP), on page 238, for more usage information.

| **remote setl2tpclient** <*TunnelName*><*remoteName*> |
|---|

## REMOTE SETLNS

This command is specific to L2TP tunnel configuration. Refer to the L2TP commands section (page 106) for usage information.

**remote setLNS** <*TunnelName*><*remoteName*>

## REMOTE SETMGMTIPADDR

This command assigns to the remote router entry an IP address which is to be used for management purposes only and not for IP address translation. This management IP address is generally a private network address used solely by the ISP.

The management IP address is separate from the IP address used for IP address translation. The IP address used for address translation is generally a public IP address valid on the Internet. It is set by the **remote setSrcIpAddr** command (page 206).

**Note:** The management address is not effective until after the next **save** and **reboot**.

**Note:** To use the management address as the source address for a ping, you must specify it using the **-I** option on the **ping** command (page 141). For example, to use management address 192.168.1.2 when pinging destination address 192.168.100.100, specify:

```
ping -I 192.168.1.2 192.168.100.100
```

**Note:** To use the management address as the source address for a copy, you must specify both the source and destination addresses on the **copy** command (page 146).

To list the current management address for the remote router, if any, use the **remote list** command (page 195). To set a management address for an Ethernet interface, see **eth ip mgmt** (page 176).

**remote setMgmtIpAddr** <*ipaddr*> <*mask*> <*remoteName*>

*ipaddr* IP address (4 decimals separated by periods).

*mask* IP subnet mask (4 decimals separated by periods).

*remoteName* Name of the remote router (character string).

**Example:**

```
remote setMgmtIpAddr 10.0.0.3 255.255.255.0 internet
```

## REMOTE SETOURPASSWD

Sets a unique CHAP or PAP authentication password for the local router that is used for authentication when the local router connects to the specified remote router. This password overrides the password set in the **system passwd** command. A common use is to set a password assigned to you by Internet Service Providers.

**remote setOurPasswd** <*password*> <*remoteName*>

*password* Authentication password of the local router for use in connecting to the remote router.

**Note:** The password is case-sensitive.

*remoteName* Name of the remote router (character string).

**Example:** `remote setOurPasswd s1dpxl7 HQ`

## REMOTE SETOURSYSNAME

Sets a unique CHAP or PAP authentication system name for the local router that is used for authentication when the local router connects to the specified remote router. This system name overrides the system name set in the **system name** command. A common use is to set a password assigned to you by Internet Service Providers.

**remote setOurSysName** <*name*> <*remoteName*>

*name* System name of the target router.

**Note:** The system name is case-sensitive and may be no more than 50 characters.

*remoteName* Name of the remote router (character string).

**Example:** `remote setOurSysName s1dpxl7 HQ`

## REMOTE SETPASSWD

Sets the CHAP or PAP authentication password that is used when the remote router establishes a connection or is challenged by the target router.

**remote setPasswd** <*password*> <*remoteName*>

*password* Authentication password of the remote router. Note that the password is case-sensitive.

*remoteName* Name of the remote router (character string).

**Example:** `remote setPasswd s2dpxl7 HQ`

## REMOTE SETPROTOCOL

Sets the link protocol for the remote router.

**remote setProtocol** [PPP | PPPLLC | RFC1483 | RFC1483MER | FRF8 | RAWIP] <*remoteName*>

PPP PPP protocol with no encapsulation.

PPLLC PPP protocol with LLC SNAP encapsulation (used with frame relay internetworking units).

RFC1483 RFC 1483 protocol.

RFC1483MER RFC 1483MER (MAC Encapsulated Routing) protocol.

FRF8 This protocol implements ATM to frame relay as defined in the Frame Relay Forum FRF.8 Interworking Agreement.

RAWIP RawIP protocol.

*remoteName* Name of the remote router (character string).

**Example:** `remote setProtocol ppp fp1`

## REMOTE SETIPSLAVEPPP

If SetIPSlaveModePPP is yes, the router will accept the IP address that the remote end informs the router that it has without reguard to how the router was previously configured. If setIPSlaveModePPP is no, the router will try to use the address that it was configured for.

Normally there is no need to change the default (no) value of this option, however, in certain situations where the router is managed by another party, (as part of a managed service), it might make sense to set this value to yes to ensure that the central management site always specifies the IP address of the router.

**remote setIPslavePPP** yes|no <*remoteName*>

## REMOTE SETPVC

Specifies the PVC number for connecting to the remote router.

**remote setPVC** <*vpi number*>*<*vci number*> <*remoteName*>

*vpi number* Virtual Path ID — number that identifies the link formed by the virtual path.

*vci number* Virtual Circuit ID — number that identifies a channel within a virtual path in a DSL/ATM environment.

*remoteName* Name of the remote router (character string).

**Example:** `remote setPVC 0*38 HQ`

## REMOTE SETRMTIPADDR

Sets the WAN IP address for the remote router. This address is required only if the remote router does not support IP address negotiation under PPP (i.e., numbered mode is required, and the remote router cannot specify a WAN IP address for use during the negotiation process).

**remote setRmtIpAddr** <*ipaddr*> <*mask*> <*remoteName*>

*ipaddr* IP address of the remote router, in the format of 4 decimals separated by periods.

*mask* IP network mask of the remote router, in the format of 4 decimals separated by periods.

*remoteName* Name of the remote router (character string).

**Example:** `remote setRmtIpAddr 128.1.210.65 255.255.255.192 HQ`

## REMOTE SETSRCIPADDR

Sets the IP address for the target WAN connection to the remote router. You may set this address when the remote router requires the target and the remote WAN IP addresses to be on the same subnetwork. Another instance is to force numbered mode and to prevent the remote router from changing the target WAN IP address through IPCP address negotiation. The target WAN IP address defaults to the Ethernet LAN IP address.

**remote setSrcIpAddr** <*ipaddr*> <*mask*> <*remoteName*>

*ipaddr* Target IP address of the WAN connection to the remote router, in the format of 4 decimals separated by periods.

*mask* IP network mask, in the format of 4 decimals separated by periods.

*remoteName* Name of the remote router (character string).

**Example:** `remote setSrcIpAddr 128.1.210.151 255.255.255.192 HQ`

## REMOTE STATS

Shows the current status of the connection to the remote router, including the bandwidth and data transfer rate.

**remote stats** [<*remoteName*>]

*remoteName* Name of the remote router (character string).

**Example:** `remote stats HQ`

**Response:**

```
STATISTICS FOR <HQ>:
  Current state ......................  currently connected
  Current output bandwidth ...........  0 bps
  Current input bandwidth ............  0 bps
  Current bandwidth allocated ........  25600000 bps
  On port ATM_VC/1 ...................  0+01:02:36 (0%/0% of 25600000 bps)
  Total connect time .................  0+01:11:48
  Total bytes out ....................  15896
  Total bytes in .....................  0

STATISTICS FOR <internet>:
  Current state ......................  not connected
  Current output bandwidth ...........  0 bps
  Current input bandwidth\ ...........  0 bps
  Current bandwidth allocated ........  0 bps
  Total connect time .................  0+00:00:00
  Total bytes out ....................  0
  Total bytes in .....................  0
```

where:

Current state: connected, not connected, currently connecting, currently attempting to connect, currently closing, out of service, or not known.

Bandwidth state: idle, increasing, decreasing, decreasing hold, unknown, or idle.

## REMOTE STATSCLEAR

Allows the user to reset the statistics counter for a given remote router.

**remote statsclear** <*remoteName*>

*remoteName* Name of the remote router (character string).

**Example:** `remote statsclear HQ`

## REMOTE UNBINDIPVIRTUALROUTE

Removes a remote route from the named IP virtual routing table.

To list the remote routes, use the **remote listIProutes** command, page 196. To add a remote route, use the **remote bindIPVirtualRoute** command, page 186.

**Note:** A route change in an IP virtual routing table takes effect immediately. However, the change is lost if it is not saved before the next **reboot**.

**remote unbindIPVirtualRoute** <*ipaddr*> <*tablename*> <*remoteName*>

*ipaddr* IP address of the remote network or station (4 decimals separated by periods).

*tablename* IP virtual routing table from which the route is removed (character string).

*remoteName* Name of the remote router (character string).

**Example:**

The following command removes a route from virtual routing table FRANCISCO. The route removed is for IP address 10.1.2.0 and remote router HQ.

```
remote unbindIPVirtualRoute 10.1.2.0 FRANCISCO HQ
```

# Asymmetric Digital Subscriber Line Commands (ADSL)

The following ADSL commands are used to manage the ADSL link for an ADSL router.

## ADSL ?

Lists the supported keywords.

| **adsl ?** |
|---|

**Response:**

```
ADSL commands:
?          restart      stats        speed
```

## ADSL RESTART

Resynchronizes the modem with the CO (Central Office) equipment.

| **adsl restart** |
|---|

**Response:**

```
# 12/02/1997-12:47:46:ADSL: Idle
12/02/1997-12:47:46:ADSL: Startup initiated
12/02/1997-12:47:48:ADSL: Startup training in progress
12/02/1997-12:47:54:ADSL: Modem started successfully
12/02/1997-12:47:54:ADSL: Near Avg SQ #: 44 dB [   3]
12/02/1997-12:47:54:ADSL: Far  Avg SQ #: 44 dB [   3]
12/02/1997-12:47:54:ADSL: Downstream rate: 6272 Kb/s, Upstream rate:
1088 Kb/s
12/02/1997-12:47:54:DOD: connecting to internet @ 0*38 over ATM_VC/1
12/02/1997-12:47:56:ADSL: Data Mode
DUM: BR CHG ATM_VC/1 - to internet now forwarding
```

## ADSL SPEED

Displays the current downstream and upstream rates.The actual speed is set by the DSLAM.

| **adsl speed** |
|---|

**Example:** `adsl speed`

**Response:**

```
downstream rate: 6272 Kb/s, upstream rate: 1088 Kb/s
```

## ADSL STATS

Shows the current error status for the ADSL connection.

| **adsl stats** [clear] |
|---|

clear Option used to reset the counters.

**Example:** `adsl stats`

**Response:**

```
ASDL Statistics:
  Out of frame errors .....  0
  HEC errors received .....  0
  CRC errors received .....  0
  FEBE errors received ....  0
Remote Out-of-frame .......  0
Remote HEC errors .........  0
```

# Asynchronous Transfer Mode Commands (ATM)

The following ATM commands are used to manage the ATM link for an ATM router.

## ATM ?

Lists the supported keywords.

| **atm ?** |
|---|

**Example:** `atm ?`

**Response:**

```
ATM commands:
?               echoPVC          findPVC
help            voicePVC
```

## ATM PCR

Sets the speed of the ATM link in cells per second. This command is similar to **atm speed** (speed in kilobytes). Refer to the command **atm speed**.

| **atm pcr** <*cells/seconds*> |
|---|

*cells/second* number of cells per second

**Example:** `atm pcr 471`

## ATM RESET

Performs traffic shaping. It causes the ATM link to re-initialize.

| **atm reset** |
|---|

**Example:** `atm reset`

## ATM SAVE

Saves the ATM configuration settings.

| **atm save** |
|---|

**Example:** `atm save`

## ATM SPEED

Sets the speed of the ATM link in kilobits per second. This command is similar to **atm pcr** (speed in cells per second). Refer to the command **atm pcr**.
The upstream speed default is 326 Kb/s. Use this command if the upstream speed exceeds 326 Kb/s. The speed value is generally obtained from your Network Service Provider.

| **atm speed** [*upstream speed in Kb/S*] |
|---|

*upstream speed in Kb/S* Number provided by the Network Service Provider. The default value for the upstream speed is 326 Kb/s.

**Example:**
```
atm speed 326
```

**Response:**
```
ATM Upstream Rate:   326 Kb/S
```

## REMOTE SETATMTRAFFIC

Sets ATM traffic-shaping on a remote router. ATM traffic-shaping allows the user to set the average rate at which cells are sent , the Sustained Cell Rate (SCR) to a value lower than the ATM link speed, the Peak Cell Rate(PCR).

**Note 1:** This command can only apply to one remote router. ATM traffic-shaping must be used if more than one remote router is defined.

**Note 2:** ATM traffic-shaping can be disabled with the command **remote setATMTraffic 0 0** <*remoteName*>

| **remote setATMTraffic** SCR MBS <*remoteName*> |
|---|

SCR Sustained Cell Rate (cells per second).

MBS Maximum Burst Size (cells).

*remoteName* Name of the remote router (character string).

**Example:** Assume that the ATM link speed (upstream) is 200 Kb/s 471 cells/s and an average upstream data rate of 20 Kb/s (47 cells/s) is desired, then you would issue the following command :
```
remote setATMtraffic 47 31 HQ
```

To disable ATM traffic-shaping on HQ, use:
```
remote setATMtraffic 0 0 HQ
```

# DMT Commands

The command manages the ADSL DMT router. To see additional DMT debug commands, see ADSL DMT Router Debug Commands, on page 297.

## DMT MODE

The **dmt mode** command can request one of three modes: ANSI, no_Trellis_ANSI, and UAWG.

UAWG mode is becoming obsolete.

No Trellis encoding for T1.413 ANSI ADSL is only needed where auto-negotiation is not supported for Trellis.

| **dmt mode ansi \| no_trellis_ansi \| uawg** |
|---|

# Dual-Ethernet Router Commands (ETH)

The following Ethernet commands are used to manage the Ethernet interfaces for the Dual-Ethernet (Ethernet-to-Ethernet) router and thus are *specific to this type of router only*.

**Note:** For non-specific Ethernet commands, refer to *Target Router Ethernet LAN Bridging and Routing (ETH), on page 167*.

## General information

This Dual-Ethernet router may be configured via the Web Browser GUI or from the Command Line Interface (CLI). You will need to use the CLI to set up any DHCP options as well as to configure optional features like IP filtering.

The Dual-Ethernet router has two interfaces:

ETH/0 = Hub with four 10Base-T connectors

ETH/1 = Single 10Base-T connector

**Note 1:** For configuration information, refer to the *Customer Release Notes* provided with the Dual-Ethernet router.

**Note 2:** If you use the **Boot from Network** option from the boot menu to perform a boot code update, the boot request is sent from the ETH/0 interface only.

## ETH BR ENABLE

Enables bridging in a Dual-Ethernet environment. This command requires rebooting the router for the change to take effect.

| **eth br enable** |
|---|

**Example:** `eth br enable`

## ETH BR DISABLE

Disables bridging in a Dual-Ethernet environment.

**Note:** This command requires rebooting the router for the change to take effect.

| **eth br disable** |
|---|

**Example:** `eth br disable`

## ETH IP ADDHOSTMAPPING

Remaps a range of local LAN IP addresses to a range of public IP addresses on a *per-interface basis*. These local addresses are mapped one-to-one to the public addresses.

**Note:** The range of public IP addresses is defined by *<first public addr>* only. The rest of the range is computed automatically (from *<first public addr>* to *<first public addr>* + number of addresses remapped - 1) inclusive.

**eth ip addHostMapping** *<first private addr><second private addr><first public addr><port#>>*

| | |
|---|---|
| *first private addr* | First IP address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods. |
| *second private addr* | Last address in the range of local IP address to be remapped, in the format of 4 decimals separated by periods. |
| *first public addr* | Defines the range of public IP addresses, in the format of 4 decimals separated by periods. The rest of the range is computed automatically. |
| *port#* | Ethernet interface number; can be 0 or 1. |
| **Example:** | `eth ip addHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 1` |

## ETH IP ADDSERVER

This Network Address Translation (NAT) command is used to add a server's IP address (on the LAN) associated with this interface for a particular protocol.

**eth ip addServer** *<ipaddr>*|discard|me *<protocolid>* |tcp|udp*<first port>* |ftp|telnet|smtp|snmp|http [*<last port>* [*<first private port>*]] *<port#>*

| | |
|---|---|
| *ipaddr* | IP address of the host selected as server in the format of 4 decimals separated by periods |
| discard | Discards the incoming server request. |
| me | Sends the incoming server request to the local router, regardless of its IP address. |
| *protocolid* | Protocol used by the selected server; can be **tcp** or **udp.** |
| *first port* | First or only port as seen by the Ethernet interface**.** Port used by the selected server. Can be a string such as ftp, telnet, smtp, snmp, or http, or a numeric value between 0 and 65,535. A numeric value of 0 will match any port |
| *last port* | If specified, this is used with *<first port>* to specify a range of ports as seen by Ethernet interface end for the server on the LAN. |
| *first private port* | If specified, this is a port remapping of the incoming request from the Ethernet interface. |
| *port#* | Ethernet interface number; can be 0 or 1. |
| **Example:** | `eth ip addServer 192.168.1.5 tcp smtp 1`<br>`eth ip addServer 192.168.1.10 tcp 9000 9000 telnet 0` |

## ETH IP DELHOSTMAPPING

Undoes an IP address/ host translation (remapping) range that was previously established with the command **eth ip addHostMapping** on a *per-interface basis*.

**eth ip delHostMapping** <*first private addr*> <*second private addr*> <*first public addr*> <*port#*>

| | |
|---|---|
| *first private addr* | First IP address in the range of IP address, in the format of 4 decimals separated by periods. |
| *second private addr* | Last address in the range of IP address, in the format of 4 decimals separated by periods. |
| *first public addr* | Defines the range of public IP addresses, in the format of 4 decimals separated by periods. The rest of the range is computed automatically. |
| *port#* | Ethernet interface number; can be 0 or 1. |
| **Example:** | `eth ip delHostMapping 192.168.207.40 192.168.207.49 10.0.20.11 1` |

## ETH IP DELSERVER

This Network Address Translation (NAT) command is used to delete an entry created by the **eth ip addServer** command.

**eth ip delServer** <*ipaddr*>|discard|me <*protocolid*> |tcp|udp <*first port*> |ftp|telnet|smtp|snmp|http [<*last port*> [<*first private port*>]] <*port#*>

| | |
|---|---|
| *ipaddr* | IP address of the host selected as server in the format of 4 decimals separated by periods |
| *discard* | Discards the incoming server request. |
| *me* | Sends the incoming server request to the local router, regardless of its IP address. |
| *protocolid* | Protocol used by the selected server; can be **tcp** or **udp.** |
| *first port* | First or only port as seen by the Ethernet interface**.** Port used by the selected server. Can be as string such as ftp, telnet, smtp, snmp, or http, or a numeric value between 0 and 65,535. A numeric value of 0 will match any port |
| *last port* | If specified, this is used with <*first port*> to specify a range of ports as seen by the Ethernet interface for the server on the LAN. |
| *first private port* | If specified, this is a port remapping of the incoming request from the Ethernet interface. |
| *port#* | Ethernet interface number; can be 0 or 1. |
| **Example:** | `eth ip delServer 192.168.1.5 tcp ftp 0` |

## ETH IP TRANSLATE

This command is used to control Network Address Translation on a *per-interface basis*. It allows several PCs to share a single IP address to the Internet.

| **eth ip translate onloff** <*port#*> |
|---|

*port#* Ethernet interface number. Can be 0 or 1.

**Example:** `eth ip translate on 0`

# High-Speed Digital Subscriber Line Commands (HDSL)

The following HDSL commands are used to manage the HDSL link for an HDSL router.

## General Information about HDSL

- ### Line activation

  Line activation is independent of network settings. During activation, the Link light (on the front panel of the router) first is yellow and then turns green when the link becomes active.

  The router at the CPE end will try auto-speed detection, starting at 384 and then try to detect the next higher speed (for about 30 seconds per speed). The WAN light should turn yellow, then green, when the link has activated.

  Auto-speed detection can be turned off with the command **hdsl speed noauto**.

  If the line was previously set to "no auto-speed" (noauto), the Link light will be amber instead, when the line tries to activate.

  The **ifs** command displays the Link as either off or opened when successfully activated. Following is a sample output.

  **Sample:**

```
ifs
Interface      Speed      In %     Out %    Protocol     State
Connection
ETHERNET/0     10.0mb     0%/0%    0%/0%    (Ethernet)   OPENED
HDSL/0         384kb      0%/0%    0%/0%    (HDSL)       OPENED
CONSOLE/0      9600 b     0%/0%    0%/0%    (TTY)        OPENED
```

- ### Auto-speed sequence

  Auto-speed starts with the lower speed (384) and then tries to activate for 30 seconds. If no activation takes place, it attempts the next higher speed. The time intervals between activation may change if the modems don't activate as expected. Following is a correct activation output.

```
03/09/1998-17:11:59:HDSL: Deactivated
03/09/1998-17:12:22:HDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:13:00:HDSL: Deactivated
03/09/1998-17:13:01:HDSL: CPE is Activating at 1168 Kb/s
03/09/1998-17:13:32:HDSL: Deactivated
03/09/1998-17:13:32:HDSL: CPE is Activating at 1168 Kb/s
03/09/1998-17:14:11:HDSL: Deactivated
03/09/1998-17:14:12:HDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:14:51:HDSL: Activated
03/09/1998-17:14:53:FRAMER: The framer is synchronized
```

## HDSL ?

Lists the supported keywords.

| **hdsl ?** |
|---|

**Example:** `hdsl ?`

**Response:**

```
HDSL commands:
?               help          terminal
save            speed
```

## HDSL SPEED

CO end: Sets the speed manually on the Central Office (CO) end only.

CPE end: The router on the Customer Premises End (CPE) is always in auto-speed mode: it uses an auto-speed algorithm to attempt to match the CO speed. The command **hdsl speed noauto** is used to override auto-speed.

**Note 1:** The command **hdsl speed** (with no option) displays the current speed if the modem has activated successfully.

**Note 2: hdsl speed noauto** should be followed by the command **hdsl save** to be persistent across reboots.

**Note 3:** During auto-speed search, use the command **hdsl speed <***speed***>** to stop the search and restart it at the speed you just entered.

| **hdsl speed** [384 \| 1168 \| noauto] |
|---|

| | |
|---|---|
| 384 | Default speed for the CO. |
| 1168 | Authorized non-default speeds for the CO in Mbps. |
| noauto | Used to override auto-speed on the CPE. |

**Example:**

```
hdsl speed 1168
hdsl speed noauto
hdsl speed
```

## HDSL SAVE

Saves the HDSL-related changes across reboots.

| **hdsl save** |
|---|

**Example:** `hdsl save`

## HDSL TERMINAL

The router is by default configured as the Customer Premises Equipment (CPE). Use this command if you intend to configure the router as the Central Office equipment (CO).

**hdsl terminal cpe** defines the CPE end (default configuration)

**hdsl terminal co** defines the CO end.

**hdsl terminal** displays the current settings.

| **hdsl terminal** [cpe\|co] |
|---|

co This option lets you define the router as the CO.

**Example:** `hdsl terminal`

**Response:**
```
Customer Premises
```

**Example:** `hdsl terminal co`

# ISDN Digital Subscriber Line (IDSL)

## General Information about IDSL

- ### Data Link Connection Identifier (DLCI)

  The IDSL router can support several DLCI virtual circuits over a Frame-Relay IDSL link. However, a typical connection to the Internet will require only one DLCI. The DLCI number must match the DLCI of the remote end. An activated router should show all green lights for LINE, CH1, CH2, and NT1 LEDs. The following IDSL commands are used to manage the IDSL link for an IDSL router.

## ISDN ?

Lists the supported keywords.

| **isdn ?** |
|---|

**Example:** `isdn ?`

**Response:**

```
ISDN commands:
?               help
save            set           list
```

## ISDN LIST

Lists the current switch type information.

| **isdn list** |
|---|

**Example:** `isdn list`

**Response:**

```
Switch type is Frame Relay IDSL 144k
```

## ISDN SAVE

Saves the IDSL-related changes across reboots.

| **isdn save** |
|---|

**Example:** `isdn save`

## ISDN SET SWITCH

Specifies link speeds of 64, 128, or 144 Kbps for the IDSL connection.

| **isdn set switch** [FR64 \| FR128 \| FR144] |
|---|

FR64 Link speed of 64 Kbps

FR128 Link speed of 128 Kbps

FR144 Link speed of 144 Kbps

**Example:** `isdn set switch fr144`

## REMOTE SETDLCI

This command allows the user to set the Data Link Connection Identifier—an address identifying a logical connection—in a Frame-Relay environment. This number is generally provided by the Network Service Provider.

| **remote setDLCI** <*dlcinumber*> <*remoteName*> |
|---|

*dlcinumber* Frame-Relay number identifying the data-link connection.

*remoteName* Name of the remote router (character string).

**Example:** `remote setDLCI 16 HQ`

## REMOTE SETPROTOCOL

This IDSL-specific command is used to select the appropriate link protocol for your IDSL connection. Your Network Service Provider will tell you which link protocol to use.

| **remote setProtocol** [PPP \| FR \| MER] <*remoteName*> |
|---|

PPP PPP protocol with no encapsulation.

FR RFC 1490 protocol (Multiprotocol encapsulation over Frame Relay).

MER RFC 1490 protocol with MAC Encapsulated Routing.

*remoteName* Name of the remote router (character string).

**Example:** remote setProtocol FR HQ

# SDSL Commands

The commands in this section can manage the Symmetric Digital Subscriber Line (SDSL) link for an SDSL router.

- **sdsl speed** Displays and sets the line speed.
- **sdsl stats** Displays and clears SDSL statistics.
- **sdsl terminal** Redefines the router as CO equipment.
- **sdsl eoc** Enables the embedded operations channel (EOC) if it is available in the router.

## Line activation

Line activation is independent of network settings. During activation, the Link light (on the front panel of the router) first is yellow and then turns green when the link becomes active. The WAN light should turn yellow, then green, when the link has activated.

If auto-speed detection was turned off for the line (**sdsl speed noauto**), the Link light is amber when the line tries to activate.

The **ifs** command displays the Link as either off or opened when it has successfully activated. The following is a sample output.

**Sample:**

```
ifs
Interface      Speed      In %     Out %     Protocol    State
Connection
ETHERNET/0     10.0mb     0%/0%    0%/0%    (Ethernet)   OPENED
SDSL/0         384kb      0%/0%    0%/0%    (ATM)        OPENED
CONSOLE/0      9600 b     0%/0%    0%/0%    (TTY)        OPENED
```

## Line speed

In general, the line activates at the speed it was last activated. The line speed can be changed by several means depending on the router model. One way is by setting the speed manually with an **sdsl speed** command.

In general, if the speed is not set otherwise, the router at the CPE end will try auto-speed detection (unless it has been disabled).

Auto-speed detection attempts to activate the line at different speeds (for about 30 seconds per speed) until the line is activated. The following is output from a successful activation.

```
03/09/1998-17:11:59:SDSL: Deactivated
03/09/1998-17:12:22:SDSL: CPE is Activating at 768 Kb/s
03/09/1998-17:13:00:SDSL: Deactivated
03/09/1998-17:13:01:SDSL: CPE is Activating at 1152 Kb/s
03/09/1998-17:13:32:SDSL: Deactivated
03/09/1998-17:13:32:SDSL: CPE is Activating at 1152 Kb/s
03/09/1998-17:14:11:SDSL: Deactivated
03/09/1998-17:14:12:SDSL: CPE is Activating at 384 Kb/s
03/09/1998-17:14:51:SDSL: Activated
03/09/1998-17:14:53:FRAMER: The framer is synchronized
```

```
03/09/1998-17:15:19:DOD: connecting to co @ 0*38 over ATM-VC/1
03/09/1998-17:15:35:DOD: link to co over ATM-VC/1 is now up
03/09/1998-17:15:57:SDSL: Line Rate at last activation saved
```

## SDSL ?

Lists the supported keywords for the **sdsl** command.

| **sdsl ?** |
|---|

**Example:**

```
# sdsl ?
SDSL commands:
?                   help                speed
save                stats               terminal
```

## SDSL EOC

Manages the embedded operations channel (EOC) if it is available in the router. To determine if this command is available, enter an **sdsl ?** command.

The EOC can be used by the central office (CO) to change the speed of the SDSL line. To be effective, the EOC must be enabled by both the CO and the customer premises equipment (CPE).

To determine if the EOC is enabled, enter the command **sdsl eoc**. To enable the EOC, enter the command **sdsl eoc enable**. (See the example at the end of this command description.)

If the EOC is enabled for the router, its SDSL line initially activates at 192 kb/s. The CO can then use the EOC to set the correct line speed.

**Note:** If you enable the EOC, you cannot manually set the line speed using the **sdsl speed** command.

**Note:** If the router has been defined as Central Office (CO) equipment, you cannot enable the EOC (see the **sdsl terminal** command on page 226).

**Note:** Remember to enter an **sdsl save** or **save** command to save SDSL changes across reboots.

| **sdsl eoc** [enable \| disable] |
|---|

enable — Enables the EOC.

disable — Disables the EOC.

**Example:** The commands in the example do the following tasks:

1. Determines that the EOC command is available.
2. Determines that the EOC is currently disabled.
3. Enables the EOC.
4. Saves the SDSL change.
5. Displays the current line speed. (The line speed is controlled by the CO; the **sdsl speed** command cannot change the line speed while the EOC is enabled.)
6. Lists the SDSL statistics. Because the EOC is enabled, the list includes EOC statistics.

```
# sdsl ?
```

```
SDSL commands:
?                   help                eoc
save                speed               stats
terminal
# sdsl eoc
EOC disabled
# sdsl eoc enable
EOC enabled
# sdsl save
# sdsl speed
SDSL Current Speed (CO-controlled): 192 Kb/s
# sdsl stats
FRAMER Statistics:
  Framer Interrupts...... 1217
  Out of frame errors.... 0
  HEC errors received.... 0
  CRC errors received.... 0
  FEBE errors received... 0
  Remote Out-of-frame.... 7
  Remote HEC errors...... 0
  EOC enabled

EOC Statistics:
  Receive EOC interrupts. 1210
  Good frames received... 0
  Overruns received ..... 0
  Aborts received ....... 0
  Illegal frames received 0
  CRC errors received ... 0
  Frames too long........ 0
  Bad frame sizes........ 0
  Unrecognized protocols. 0
  Wrong EOC Versions..... 0
  Transmit EOC interrupts 1210
  Frames transmitted..... 0
```

## SDSL SAVE

Saves SDSL configuration changes across reboots.

| **sdsl save** |
|---|

**Example:** `sdsl save`

## SDSL SPEED

Manages the speed of the SDSL line.

At the Central Office (CO) end, the command sets the speed manually only.

At the Customer Premises Equipment (CPE) end, the command can:

- Display the current speed setting and list the available speeds (**sdsl speed**)

- Manually set the speed (**sdsl speed <*speed*>**)
- Override auto-speed detection (**sdsl speed noauto**)

**Note:** To re-instate auto-speed detection, enter an **sdsl speed** <*speed*> command.

**Note:** If the EOC is enabled, the line speed is controlled by the CO and cannot be changed by an **sdsl speed** command. (See the **sdsl eoc** command on page 223.)

**Note:** Remember to enter an **sdsl save** or **save** command to save SDSL changes across reboots.

| **sdsl speed** [<*speed*> | noauto] |
|---|

*speed* — Speed in kbps. To see the speeds available for the model type, enter **sdsl speed**. If the auto-speed search is in progress, this command stops the search and sets the line speed as specified on the command.

noauto — Overrides auto-speed detection. If auto-speed detection is disabled, the Link light on the front panel is amber when the line tries to activate.

(Auto-speed detection is reinstated if you enter an **sdsl speed** <*speed*> command.)

**Example:**

The example shows three commands:

1. Displays the current line speed, indicates that the line speed is set by auto-speed detection, and lists the available speed options.
2. Requests a line speed of 1152 Kb/s.
3. Shows that the line speed has been changed and auto-speed detection is no longer in effect.

```
# sdsl speed
SDSL Current Speed: [AUTO] 768 Kb/s
usage: sdsl speed <value in Kb/s> [ 192 384 768 1152 1536 ] | noauto
# sdsl speed 1152
# sdsl speed
SDSL Current Speed: 1152 Kb/s
usage: sdsl speed <value in Kb/s> [ 192 384 768 1152 1536 ] | noauto
```

## SDSL STATS

Displays SDSL frame statistics. It can also clear the SDSL statistic counters.

| **sdsl stats** [clear] |
|---|

clear — Clears all SDSL statistics counters.

**Example:**

```
# sdsl stats
FRAMER Statistics:
  Framer Interrupts..... 2118
  Out of frame errors.... 1
  HEC errors received.... 16
  CRC errors received.... 3
  FEBE errors received... 2
```

```
Remote Out-of-frame.... 16
Remote HEC errors...... 0
EOC disabled
```

**Note:** For an SDSL statistics example with the EOC enabled, see the **sdsl eoc** command, page 223.

## SDSL TERMINAL

Displays and/or changes the router's status as CO or CPE.

The router is, by default, configured as Customer Premises Equipment (CPE). Use this command if you intend to configure the router as Central Office equipment (CO).

To determine the current CO/CPE setting, enter **sdsl terminal**.

| **sdsl terminal** [cpe \| co] |
|---|

cpe Defines the router as the customer premises (CPE) equipment.

co Defines the router as the central office (CO) equipment.

**Example:**

```
# sdsl terminal
Customer Premises
```

Dynamic Host Configuration Protocol Commands (DHCP)The following DHCP commands allow you to disable and disable subnetworks and client leases.

# DHCP (Dynamic Host Configuration Protocol) Commands

The following DHCP commands allow you to:

- Enable and disable subnetworks and client leases.
- Add subnetworks and client leases.
- Set the lease time.
- Change client leases manually.
- Set option values globally, for a subnetwork, or for a client lease.
- Enable/disable BootP.
- Use BootP to specify the boot server.
- Define option types.

To read about DHCP concepts and the DHCP configuration process, see DHCP (Dynamic Host Configuration Protocol), on page 81.

## DHCP ?

Lists the supported keywords.

| **dhcp ?** |
|---|

**Response:**

```
Sub-commands for dhcp
?                   help                set
list                bootp               clear
enable              add                 del
disable             relay
```

## DHCP ADD

Adds a subnetwork, a client lease, or an option type.

| **dhcp add** [<*net*> <*mask*> ]| <*ipaddr*> | <*code*><*min*><*max*><*type*> |
|---|

| | |
|---|---|
| *net* | IP address of the subnetwork lease in the format of 4 decimals separated by periods. |
| *mask* | IP network mask, in the format of 4 decimals separated by periods. |
| *ipaddr* | IP address of the client lease in the format of 4 decimals separated by periods. |
| *code* | The user-defined code can be a number between 128 to 254 or a keyword. |

*min* Minimum number of value(s).

*max* Maximum number of value(s).

*type* Byte | word | long | longint | binary | ipaddress | string

**Example 1:** `dhcp add 192.168.254.0.255.255.255.0`
(adds this subnetwork)

**Example 2:** `dhcp add 192.168.254.31`
(adds this client lease

**Example 3:** `dhcp add 128 1 4 ipAddress`
(adds this option type

**Note:** In example 3, 128 allows IP addresses, the server has a minimum of one IP address, the server can have up to four IP addresses, and the type is “ipaddress”).

## DHCP BOOTP ALLOW

Allows a BootP request to be processed for a particular client or subnet.

**dhcp bootp allow** <*net*>|<*ipaddr*>

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Example:** `dhcp bootp allow 192.168.254.0`

## DHCP BOOTP DISALLOW

Denies processing of  a BootP request for a particular client or subnet.

**dhcp bootp disallow** <*net*>|<*ipaddr*>

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Example:** `dhcp bootp disallow 192.168.254.0`

## DHCP BOOTP FILE

Specifies the boot file name (kernel).

**Note:** Be sure to the TFTP server IP address when you specify the file.

**dhcp bootp file** [<*net*>|<*ipaddr*>]<*name*>

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*name* Name of the file to boot from; the default name for this file is KERNEL.F2K.

**Example:** `dhcp bootp file 192.168.254.0 Kernel.f2k`

## DHCP BOOTP TFTPSERVER

Specifies the TFTP server (boot server).

**dhcp bootp tftpserver** [<*net*>|<*ipaddr*>]<*tftpserver ipaddr*>

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*tftpserver ipaddr* IP address of the TFTP server in the format of 4 decimals separated by periods. To clear the IP address of the server, use 0.0.0.0.

**Examples:**
```
dhcp bootp tftpserver 192.168.254.7
dhcp bootp tftpserver 192.168.254.0 192.168.254.8
dhcp bootp tftpserver 192.168.254.21 192.168.254.9
dhcp bootp tftpserver 0.0.0.0
```

## DHCP CLEAR ADDRESSES

Clears the values from a pool of addresses.

**dhcp clear addresses** <*net*>

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

**Example:** `dhcp clear addresses 192.168.254.0`

## DHCP CLEAR EXPIRE

Releases the client lease. It then becomes available for other assignments.

**dhcp clear expire** <*ipaddr*>

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Example:** `dhcp clear expire 192.168.254.12`

**Note:** The client does not get updated;  it will still have the old value.

## DHCP CLEAR VALUEOPTION

Clears the value for a global option, for an option associated with a subnetwork, or with a specific client.

| **dhcp clear valueoption** [<*net*>|<*ipaddr*>] <*code*> |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*code* Code can be a number between 1 and 61 or a keyword. Use the command **dhcp list definedoptions** to list the codes and keywords.

**Examples:**
```
dhcp clear valueoption 4
dhcp clear valueoption 192.168.254.0 7
dhcp clear valueoption 192.168.254.2 gateway
```

## DHCP DEL

Deletes a subnetwork lease, a specific client lease, or a code.

| **dhcp del** <*net* |<*ipaddr*>|<*code*> |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*code* The user-defined code and can be a number between 128 to 254 or a keyword.

**Example 1:**
```
dhcp del 192.168.254.0
```
(deletes this subnetwork)

**Example 2:**
```
dhcp del 192.168.254.31
```
(deletes this client lease

**Example 3:**
```
dhcp del 128
```
(deletes this option with code 128)

## DHCP DISABLE

Disables a subnetwork or a client lease.

| **dhcp disable** all | <*net*> | <*ipaddr*> |
|---|

all Disables all subnets.

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Examples:**
```
dhcp disable 192.168.254.0
dhcp disable 192.168.254.17
```

## DHCP ENABLE

Enables a subnetwork or a client lease.

| **dhcp enable** all \| <*net*>\|<*ipaddr*> |
|---|

all Enables all subnets.

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Examples:**
```
dhcp enable 192.168.254.0
dhcp enable 192.168.254.17
```

## DHCP LIST

Lists global, subnetwork, and client lease information.

| **dhcp list** \| <*net*>\|<*ipaddr*> |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

**Example 1:** To list global information, enter:
```
dhcp list
```

**Response:**
```
       bootp server ..............      none
       bootp file ................
       DOMAINNAMESERVER (6) ......      192.168.210.20 192.84.210.21
       DOMAINNAME (15) ...........      flowpoint.com
       WINSSERVER (44) ...........      192.168.254.73
Subnet 192.168.254.0, Enabled
       Mask ......................      255.255.255.0
       first ip address ..........      192.168.254.2
       last ip address ...........      192.168.254.253
       lease .....................      Default
       bootp .....................      not allowed
       bootp server ..............      none
       bootp file ................
    GATEWAY (3) ..................      192.168.254.254
   client 192.168.254.2, Ena, jo-computer, Expired
   client 192.168.254.3, Ena, Jo, 1999/5/16 11:31:33
```

**Example 2:** To list information for client 192.168.254.3, enter
`dhcp list 192.168.254.3`

**Response:**

```
Client 192.168.254.3, Enabled
    lease ..........................  Default
    expires ........................  1998/5/16 11:31:33
    bootp ..........................  not allowed
    bootp server ...................  none
    bootp file .....................
    HOSTNAME (12) ..................  JO
    CLIENTIDENTIFIER (61) ..........  1 2 96 140 76 149 180
```

**Example 3:** To list information for the subnetwork 192.168.254.0, enter:
`dhcp list 192.168.254.0`

**Response:**

```
Subnet 192.168.254.0, Enabled
    Mask............................  255.255.255.0
    first ip address................  192.168.254.2
    last ip address.................  192.168.254.253
    lease...........................  Default
    bootp...........................  not allowed
    bootp server....................  none
    bootp file......................
    GATEWAY (3).....................  192.168.254.254
client 192.168.254.2, Ena, Jo-computer, Expired
client 192.168.254.3, Ena, Jo, 1998/5/16 11:31:33
```

## DHCP LIST DEFINEDOPTIONS

Lists all available predefined and user-defined options.

**Note:** For description of the predefined options listed below, refer to RFC 1533. A predefined code can be a number between 1 and 61 or a keyword. A user-defined code can be a number between 128 and 254 or a keyword.

| **dhcp list definedoptions** \| <*code*> \| <*string*> |
|---|

*code* Predefined or user-defined number or keyword.

*string* Character string.

**Example 1:** To list all available options (they may be predefined as in the list below, and/or user-defined), enter:
`dhcp list definedoptions`

**Response:**

```
code TIMEOFFSET (2), 1 occurrence, type LONG
code GATEWAY (3), 1 to 63 occurrences, type IPADDRESS
code TIMESERVER (4), 1 to 63 occurrences, type IPADDRESS
code NAMESERVER (5), 1 to 63 occurrences, type IPADDRESS
code DOMAINNAMESERVER code SUBNETMASK (1), 1 occurrence, type IPADDRESS-RESERVED
 (6), 1 to 63 occurrences, type IPADDRESS
code LOGSERVER (7), 1 to 63 occurrences, type IPADDRESS
code COOKIESERVER (8), 1 to 63 occurrences, type IPADDRESS
code LPRSERVER (9), 1 to 63 occurrences, type IPADDRESS
code IMPRESSSERVER (10), 1 to 63 occurrences, type IPADDRESS
code RESOURCELOCATION (11), 1 to 63 occurrences, type IPADDRESS
code HOSTNAME (12), 1 to 255 characters, type STRING
code BOOTFILESIZE (13), 1 occurrence, type WORD
code MERITDUMPFILE (14), 1 to 255 characters, type STRING
code DOMAINNAME (15), 1 to 255 characters, type STRING
code SWAPSERVER (16), 1 occurrence, type IPADDRESS
code ROOTPATH (17), 1 to 255 characters, type STRING
code EXTENSIONSPATH (18), 1 to 255 characters, type STRING
code IPFORWARDING (19), 1 occurrence, type BINARY
code NONCALSOURCERTE (20), 1 occurrence, type BINARY
code POLICYFILTER (21), 1 to 31 occurrences, type IPADDRESS
code MAXDGMREASSEMBLY (22), 1 occurrence, type WORD
code DEFAULTIPTTL (23), 1 occurrence, type BYTE
code PATHMTUAGETMOUT (24), 1 occurrence, type LONGINT
code PATHMTUPLATEAUTBL (25), 1 to 127 occurrences, type WORD
code INTERFACEMTU (26), 1 occurrence, type WORD
code ALLSUBNETSLOCAL (27), 1 occurrence, type BINARY
code BROADCASTADDRESS (28), 1 occurrence, type IPADDRESScode PERFORMMASKDSCVR
(29), 1 occurrence, type BINARY
code MASKSUPPLIER (30), 1 occurrence, type BINARY
code PERFORMRTRDSCVR (31), 1 occurrence, type BINARY
code RTRSOLICITADDR (32), 1 occurrence, type IPADDRESS
code STATICROUTE (33), 1 to 31 occurrences, type IPADDRESS
code TRAILERENCAP (34), 1 occurrence, type BINARY
code ARPCACHETIMEOUT (35), 1 occurrence, type LONGINT
code ETHERNETENCAP (36), 1 occurrence, type BINARY
code TCPDEFAULTTTL (37), 1 occurrence, type BYTE
code TCPKEEPALIVEINTVL (38), 1 occurrence, type LONGINT
code TCPKEEPALIVEGARBG (39), 1 occurrence, type BINARY
code NETINFOSVCDOMAIN (40), 1 to 255 characters, type STRING
code NETINFOSERVERS (41), 1 occurrence, type IPADDRESS
code NETTIMEPROTOSRVRS (42), 1 occurrence, type IPADDRESS
code VENDORSPECIFIC (43), 1 to 255 occurrences, type BYTE
code WINSSERVER (44), 1 to 63 occurrences, type IPADDRESS
code NETBIOSTCPDGMDIST (45), 1 to 63 occurrences, type IPADDRESS
code NETBIOSTCPNODETYP (46), 1 occurrence, type BYTE
code NETBIOSTCPSCOPE (47), 1 to 255 characters, type STRING
code XWSFONTSERVER (48), 1 to 63 occurrences, type IPADDRESS
code XWSDISPLAYMANAGER (49), 1 to 63 occurrences, type IPADDRESS
code REQUESTEDIPADDR (50), 1 occurrence, type IPADDRESS-RESERVED
code IPADDRLEASETIME (51), 1 occurrence, type LONGINT-RESERVED
code OPTIONOVERLOAD (52), 1 occurrence, type BYTE-RESERVED
```

```
code MESSAGETYPE (53), 1 occurrence, type BYTE-RESERVED
code SERVERIDENTIFIER (54), 1 occurrence, type IPADDRESS-RESERVED
code PARAMREQUESTLIST (55), 1 to 255 occurrences, type BYTE-RESERVED
code MESSAGE (56), 1 to 255 characters, type STRING-RESERVED
code MAXDHCPMSGSIZE (57), 1 occurrence, type WORD-RESERVED
code RENEWALTIME (58), 1 occurrence, type LONGINT
code REBINDTIME (59), 1 occurrence, type LONGINT
code CLASSIDENTIFIER (60), 1 to 255 occurrences, type BYTE
code CLIENTIDENTIFIER (61), 2 to 255 occurrences, type BYTE
code NOTDEFINED62 (62), 1 to 255 occurrences, type BYTE
code NOTDEFINED63 (63), 1 to 255 occurrences, type BYTE
code NISDOMAIN (64), 1 to 255 characters, type STRING
code NISSERVERS (65), 1 to 63 occurrences, type IPADDRESS
code TFTPSERVERNAME (66), 4 to 255 characters, type STRING
code BOOTFILENAME (67), 1 to 255 characters, type STRING
code MOBILEIPHOMEAGNT (68), 0 to 63 occurrences, type IPADDRESS
code SMTPSERVERS (69), 1 to 63 occurrences, type IPADDRESS
code POP3SERVERS (70), 1 to 63 occurrences, type IPADDRESS
code NNTPSERVERS (71), 1 to 63 occurrences, type IPADDRESS
code WWWSERVERS (72), 1 to 63 occurrences, type IPADDRESS
code FINGERSERVERS (73), 1 to 63 occurrences, type IPADDRESS
code IRCSERVERS (74), 1 to 63 occurrences, type IPADDRESS
code STREETTALKSERVERS (75), 1 to 63 occurrences, type IPADDRESS
code STREETTALKDASRVRS (76), 1 to 63 occurrences, type IPADDRESS
```

**Example 2:** TTo list options starting with the string “ga”, use:

```
dhcp list definedoptions ga
```

**Response:**

```
code,           number of values,        type of value
code GATEWAY (3), occurrence 1, type IPADDRESS
```

## DHCP LIST LEASE

Lists the lease time.

| dhcp list lease |
|---|

**Example:**

```
dhcp list lease
```

**Response:**

```
Default lease time ......... 168 hours
```

## DHCP RELAY

Lets the router relay DHCP or BootP requests to a DHCP server on the WAN when a PC attempts to acquire an IP address using DHCP. This command disables the router’s DHCP server.

| **dhcp relay** <*ipaddr*> |
|---|

*ipaddr* IP address of the target router in the format of 4 decimals separated by periods.

**Example:** `dhcp relay 128.1.210.64`

## DHCP SET ADDRESSES

Creates or changes a pool of IP addresses that are associated with a subnetwork.

| **dhcp set addresses** <*first ipaddr*> <*last ipaddr*> |
|---|

*first ipaddr* First address in a pool of addresses for a particular subnetwork.

*last ipaddr* Last address in a pool of addresses for a particular subnetwork.

**Example:** `dhcp set addresses 192.168.254.1 192.168.254.250`

## DHCP SET EXPIRE

This command is used to *manually* change a client lease expiration time to a certain value.

**Note 1:** Changing a client lease time manually is rarely required.

**Note 2:** The client information does not get updated. It will still have the old value.

| **dhcp set expire** <*ipaddr*><*hours*>\|default\|infinite |
|---|

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*hours* Lease time; minimum is 1 hour; the global default is 168 hours.

default Lease time that has been specified at the subnetwork or global level.

infinite No lease time limit; the lease becomes permanent.

**Example:** `dhcp set expire 192.168.254.18 8`

## DHCP SET LEASE

Controls lease time.

| **dhcp set lease** [<*net*>\|<*ipaddr*>]<*hours*>\|default\|infinite |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*ipaddr* IP address of the client lease in the format of 4 decimals separated by periods.

*hours* Lease time; minimum is 1 hour; the global d
efault is 168 hours.

default Lease time that has been specified at the subnetwork or global level.

infinite No lease time limit; the lease becomes permanent.

**Example 1:** `dhcp set lease 192.168.254.17 default`
(sets client lease time to default)

**Example 2:** `dhcp set lease 192.168.254.0 infinite`
(sets lease time to infinite for this subnet)

**Example 3:** `dhcp set lease 192.168.254.0 infinite`
(sets lease time to infinite for this subnet)

## DHCP SET OTHERSERVER

This command instructs the router's DHCP server to either continue or stop sending DHCP requests when another DHCP server is detected on the LAN. The default is **stop**.

| **dhcp set otherserver** <*net*> continuelstop |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

continue The router's DHCP server continues sending DHCP requests, even if another DHCP server is detected on the LAN.

stop The router's DHCP server stops sending DHCP requests when another DHCP server is detected on the LAN.

**Example:** `dhcp set otherserver 192.168.254.17 stop`

## DHCP SET MASK

Used to conveniently change the mask of a DHCP subnet without having to delete and recreate the subnet and all its entries.

| **dhcp set mask** <*net*> <*mask*> |
|---|

*net* IP address of the subnetwork lease in the format of 4 decimals separated by periods.

*mask* IP network mask, in the format of 4 decimals separated by periods.

**Example:** `dhcp set mask 192.168.254.0 255.255.255.0`

## DHCP SET VALUEOPTION

Sets values for global options, options specific to a subnetwork, or options specific to a client lease. For more information, see Setting Option Values, on page 85.

| **dhcp set valueoption** [<*ipaddr*>|<*net*>] <*code*> <*value*>.... |
|---|

*ipaddr* Specify the client IP address if the option value applies only to the client lease (4 decimals separated by periods).

*net* Specify the subnetwork IP address if the option value applies only to the subnetwork lease (4 decimals separated by periods).

*code* Code specifying the option to be set. It can be a number between 1 and 61 or a keyword. Use the command **dhcp list definedoptions** to list the codes and keywords (see DHCP LIST DEFINEDOPTIONS, on page 232.)

*value* Value to be assigned to the specified option. It could be a byte, word, signed long, unsigned long, binary, IP address, or string depending on the option.

**Example 1:** This command does not specify an client or subnetwork address, and thus sets a *global* value for the *domainnameserver* option.

```
dhcp set valueoption domainnameserver 192.168.254.2 192.168.254.3
```

**Example 2:** This command sets the value for the *gateway* option associated with the subnetwork.

```
dhcp set valueoption gateway 192.168.254.0 192.168.254.254
```

**Example 3:** This command sets a value for the *winserver* option associated with a *specific client*.

```
dhcp set valueoption 192.168.254.251 winserver 192.168.254.7
```

**Example 4:** This command sets a static route (option 33) to IP address 192.168.253.253 through router 192.168.254.254. (No mask is specified.)

```
dhcp set valueoption 33 192.168.254.254 192.168.253.253
```

# L2TP — Virtual Dial-Up Configuration (L2TP)

The following L2TP commands allow you to add, delete, and modify tunnels. L2TP router information that can be configured includes:

- Names
- Security authentication protocols and passwords
- Addresses
- Management of  traffic performance

**Note:** Two **remote** commands specific to L2TP are also included in this section.

## L2TP ?

Lists the supported keywords.

| **l2tp ?** |
|---|

**Response:**

```
L2tp Sub-commands:
?                   add                 del
forward             list                set
call                close
```

## L2TP ADD

Creates a tunnel entry.

| **l2tp add** *<TunnelName>* |
|---|

*TunnelName* Name of the tunnel (character string).  The name is case sensitive.

**Example:** `l2tp add PacingAtWork`

## L2TP SET ADDRESS

Used to define the IP address of the other end of the tunnel, either the remote L2TP Access Concentrator (LAC) or remote L2TP Network  Server (LNS).

**Caution:** If the IP address of the remote tunnel is part of a subnet that is also reached through the tunnel, a routing table entry for this address *must* be explicitly added. Normally, this routing entry will be added to remote entry, which has the default route.

**Note 1:** When a remote router tries to create a tunnel, the remote router's IP address is not authenticated .

**Note 2:** If this command is not used, then <*ipaddr*> defaults to 0.0.0.0, and this end cannot initiate the tunnel.

| **l2tp set address** <*ipaddr*> <*TunnelName*> |
|---|

*ipaddr* IP address of the remote LAC or LNS.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set address 192.168.100.1 PacingAtWork`

## L2TP SET AUTHEN

Enables or disables authentication of the remote router during tunnel establishment using the CHAP secret, if it exists. If the remote router tries to authenticate the local end during tunnel authentication, the local router will always attempt to respond, provided a CHAP secret has been configured.

| **l2tp set authen onloff** <*TunnelName*> |
|---|

**on** Enables authentication.

**off** Disables authentication.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set authen PacingAtWork`

## L2TP CALL

This command is primarily used for debugging purposes and it establishes a tunnel without creating a session.

| **l2tp call** <*TunnelName*> |
|---|

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp call PacingAtWork`

## L2TP SET CHAPSECRET

Creates a CHAP secret. This CHAP secret is used to authenticate the creation of the tunnel and is used for hiding certain control packet information. The LAC and the LNS can share a single CHAP secret for a given tunnel.

| **l2tp set CHAPSecret** <*secret*> <*TunnelName*> |
|---|

*secret* CHAP secret (character string) used to authenticate the creation of the tunnel.

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp set CHAPSecret PacingAtWork`

## L2TP CLOSE

Closes an L2TP tunnel and/or session.

| **l2tp close** <*L2TP unit number*>\|-*n*<*TunnelName*>\|-*t*<*tunnelid*>\|-*s*<*serialnum*>\|-*c*<*callid*> |
|---|

L2TP unit number

*-n TunnelName* Name of the tunnel (character string). The name is case sensitive.

*-t tunnelid* Local tunnel id.

*-s serialnum* Serial number of the call within the tunnel.

*-c callid* ID of the local call for the session.

**Note:** Either <*TunnelName*> or <*tunnelid*> must be specified.

**Example:** `l2tp close -n PacingAtWork`

## L2TP DEL

Deletes a tunnel entry.

| **l2tp del** <*TunnelName*> |
|---|

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp del PacingAtWork`

## L2TP FORWARD

The router can be configured to forward all incoming calls to an LNS without answering the incoming call. This feature is normally used when the router is acting as a LAC or both a LAC and LNS.

**Note:** Only *one* tunnel entry can have this option set.

| **l2tp forward all\|none** <*TunnelName*> |
|---|

**all** Forward all incoming call through the tunnel to an LNS

**none** No incoming calls are allowed to be forwarded through the tunnel to an LNS

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp forward PacingAtWork`

## L2TP LIST

Provides a complete display of the current configuration settings for tunnel(s), except for the authentication password/secret.

| **l2tp list** \|<*TunnelName*>\| |
|---|

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp list PacingAtWork`

```
# l2tp list
INFORMATION FOR <pacingAtWork>
  type ..........................       L2TPClient (LAC-will not dial)/LNS
All Incoming Calls Tunneled here .       no
  CHAP challenge issued ..........       yes
  hidden AVPs used ...............       yes
  sequencing/pacing ..............       window pacing
    sequencing/pacing is .........       required
    window size for sequencing/pacing    10
  ip address .....................       10.0.0.1
  Our host name ..................       pacingAtHome

  ACTIVE TUNNEL ..................       UNKNOWN
    current state ................       CLOSED
    LOCAL TUNNEL ID ..............       1
    REMOTE TUNNEL ID .............       0
    remote firmware ..............       0
    remote ip address ............       10.0.0.1
    LAC SESSION serial number ... '      0
      current state ..............       CLOSED
      LOCAL CALL ID ..............       1
        local window size ........       10
        sequencing/pacing ........       WINDOW PACING
          sequencing/pacing is ...       required
      REMOTE CALL ID .............       0
        remote window size .......       0
```

## L2TP SET DIALOUT

Lets the LNS instruct the L2TP client to use an ISDN phone line to place a call on its behalf.

| **l2tp set dialout yes\|no** <*TunnelName*> |
|---|

**yes** This option lets the router place outgoing calls.

**no** This option prevents the router from placing outgoing calls.The default is no.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set dialout yes PacingAtWork`

## L2TP SET HIDDENAVP

Configures the router to protect some L2TP control information (such as names and passwords for a PPP session) using hidden AVPs. This command is often used to turn off hidden AVPs (no option), in cases where the other end of the tunnel does not support hidden AVPs.

| **l2tp set hiddenAVP yes\|no** <*TunnelName*> |
|---|

**yes** This option lets the router hide AVPs. The default is yes.

**no** This option disables hidden AVPs.

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp set hiddenAVP yes PacingAtWork`

## L2TP SET OURPASSWORD

Specifies the router's secret/password for PPP authentication on a per-tunnel basis.

| **l2tp set ourpassword** <*password*> <*TunnelName*> |
|---|

*password* Router's secret/password used for authentication when challenged by another router.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set ourpassword PacingAtWork`

## L2TP SET OURSYSNAME

Specifies the router's name for PPP authentication on a per-tunnel basis.

| **l2tp set oursysname** <*name*> <*TunnelName*> |
|---|

*name* Name of the router that is used for authentication when challenged by another router.

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp set oursysname myName PacingAtWork`

## L2TP SET OURTUNNELNAME

This command creates local router's host name.

**Note:** If this command is not used, then, if it has been specified, the <*name*> from the **l2tp set ourSysName** command or the <*name*> from the command **system name** <*name*> is used.

| **l2tp set ourTunnelName** <*name*> <*TunnelName*> |
|---|

*name* Host name of the local router. This is the fully qualified domain name of the local router.

The name is case-sensitive

*TunnelName* Name of the tunnel (character string). The name is case sensitive.

**Example:** `l2tp set ourTunnelName isp PacingAtWork`

## L2TP SET REMOTENAME

Creates the host name of the remote tunnel.

**Note:** If this command is not used, then <*TunnelName*> of the tunnel entry is used.

| **l2tp set remoteName** <*name*> <*TunnelName*> |
|---|

*name* Host name of the remote tunnel. This is the fully qualified domain name of the remote host.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set remoteName isp PacingAtWork`

## L2TP SET TYPE

Defines the type of L2TP support for the tunnel. The router's role is defined on a per-tunnel basis.

| **l2tp set type all\|lac\|lns\|l2tpclient\|disabled** <*TunnelName*> |
|---|

**all** The router is configured to act as both a LAC/L2TP client and an LNS server.

**lac** The router is configured to act as a LAC for this tunnel.

**lns** The router is configured to act as an LNS for this tunnel.

**l2tpclient** The router is configured to act as an L2TP client for this tunnel.

**disabled** The tunnel entry is disabled.

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

**Example:** `l2tp set type l2tpclient PacingAtWork`

## L2TP SET WINDOW

Enhances traffic performance in a tunneling environment. The command's options affect the way incoming payload packets are processed. The router is configured with the following default options: sequencing, required, and size 10.

| **l2tp set window sequencing\|pacing\|nosequencing\|optional\|required\|size** <*TunnelName*> |
|---|

**sequencing** Sequence numbers are placed in the L2TP payload packets. With this option, one end instructs the other end to send sequence packets. No acknowlegments are issued for received packets.

**pacing** Sequence numbers are placed in the L2TP payload packets. When a session is created, the router specifies a window size. Acknowledgments for received packets are issued.

**nosequencing** No sequence numbers are placed in the L2TP payload packets carrying the PPP packets. If the remote end carries out sequencing or pacing, the router can still send and receive sequenced packets.

**optional** Allows dynamic switching of a session from pacing or sequencing to nosequencing.

**required** Disables dynamic switching from pacing or sequencing to nosequencing.

**size** Controls the size of the receive window for receiving packets for sequencing or pacing, when a session is created. Size can be 0 for packet sequencing. Must be a non-zero value for window pacing. Size must be less than or equal to 30.

*TunnelName* Name of the tunnel (character string). This name is case-sensitive.

**Example:** `l2tp set window sequencing PacingAtWork`

## REMOTE SETL2TPCLIENT

With this command, this remote is the path to the L2TP client and accepts tunnel calls. Use this command if your router acts as an LNS. You must also specify PPP authentication and IP routes for this remote.

| **remote setl2tpclient** <*TunnelName*><*remoteName*> |
|---|

*TunnelName* Name of the tunnel (character string) associated with the remote LAC. The name is case-sensitive.

*remoteName* Name of the remote entry (character string). The name is case sensitive.

**Example:** `remote setl2tpclient PacingAtWork Router2`

## REMOTE SETLNS

With this command, this remote is the path to the LNS, and it will forward the incoming call (which matches this remote entry) through the tunnel named <*TunnelName*> if your router is the client.

**Note:** The remote entry must also have appropriate information such as PPP authentication, IP routing, IPX routing, bridging, or Caller ID.

**remote setLNS** <*TunnelName*><*remoteName*>

*TunnelName* Name of the tunnel (character string). The name is case-sensitive.

*RemoteName* Name of the remote entry (character string).

**Example:** `remote setLNS PacingAtWork lnsServer`

# Bridge Filtering Commands (FILTER BR)

Bridge filtering allows you to control the packets transferred across the router. This feature can be used to enhance security or improve performance. Filtering is based on matched patterns within the packet at a specified offset. Two filtering modes are available.

- Deny mode will discard any packet that matches the deny filter database and let all other packets pass.
- Allow mode will only pass the packets that match the allow filter database and discard all others.

Up to 40 deny and 40 allow filters can be activated from the filter database.

## FILTER BR ?

Lists the supported keywords.

| **filter br ?** |
|---|

**Response:**

```
Bridge filter commands:
?                    add                  del
use                  list
```

## FILTER BR ADD

Adds a bridging filter to the filtering database.

| **filter br add** [*pos*] [*data*] allow | deny |
|---|

*pos* Byte offset within a packet; number from 0-127.

*data* Hexadecimal number up to 6 bytes.

**Example:** `filter br add 12 8035 deny`
(This filter prevents forwarding of RARP packets across the bridge)

## FILTER BR DEL

Deletes a bridging filter from the filtering database.

| **filter br del** [*pos*] [*data*] allow | deny |
|---|

*pos* Byte offset within a packet; number from 0-127.

*data* Hexadecimal number up to 6 bytes.

**Example:** `filter br del 12 8035 deny`

## FILTER BR LIST

Lists the bridging filters in the filtering database.

| **filter br list** |
|---|

**Example:** `filter br list`

**Response:**

```
Allow Filter:

Deny Filter:
pos:12, len=2, <80><35>
```

## FILTER BR USE

Sets the mode of filtering to either deny, allow, or none.

| **filter br use** none \| deny \| allow |
|---|

**Example:** `filter br use allow`

# Internet Key Exchange (IKE) Commands

The IKE software option and the IKE commands are described in IPSec (Internet Protocol Security), on page 119.

## IKE FLUSH

Clears all IKE configuration information from the router. For more information about IKE, see IPSec (Internet Protocol Security), on page 119.

**ike flush**

## IKE IPSEC POLICIES ADD

Defines the name of an IPsec policy to be used for filtering. Other IPSec Policy commands define the filtering parameters (see IKE IPSec Policy Commands, on page 127).

**ike ipsec policies add** <*PolicyName*>

*PolicyName* New name for an IPsec policy. To see the policy names in use, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies add mypolicy`

## IKE IPSEC POLICIES DELETE

Deletes an existing IPSec policy. To define IPSec Policies, see IKE IPSec Policy Commands, on page 127.

**ike ipsec policies delete** <*PolicyName*>

*PolicyName* Name of an existing IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies delete mypolicy`

## IKE IPSEC POLICIES DISABLE

Disables an IPSec policy. The policy can be re-enabled using the **enable** command (see IKE IPSec Policy Commands, on page 127).

**ike ipsec policies disable** <*PolicyName*>

*PolicyName* Name of an existing IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies disable mypolicy`

## IKE IPSEC POLICIES ENABLE

Enables an IPSec policy. An enable command is required for each new policy; the enable command indicates that the specification of the policy is complete and the policy is ready to be used. The enable command can also be used to re-enable a disabled policy. For more information, see IKE IPSec Policy Commands, on page 127.

**ike ipsec policies enable** <*PolicyName*>

*PolicyName* Name of the IPsec policy. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies enable mypolicy`

## IKE IPSEC POLICIES LIST

Lists the IPSec policies. For more information, see IKE IPSec Policy Commands, on page 127.

**ike ipsec policies list**

**Example:**

```
# ike ipsec policies list
IKE IPSec policies:
mypolicy (enabled)
  Source address/mask: 192.168.16.0/255.255.255.0
  Destination address/mask: 192.168.23.0/255.255.255.0
  Protocol: *
  Source port: *
  Destination port: *
  Tunnel mode
  Peer: my_aggressive_peer (0.0.0.0)
  Proposals: myproposal
```

## IKE IPSEC POLICIES SET DEST

Defines a *destination* filtering parameter value for the policy. The destination parameter requires that the data be intended for the specified destination IP address and mask. The destination is the device or network that finally receives the packet, not the router that routes the packet.

**ike ipsec policies set dest** <*IPaddress*> <*IPmask*> <*PolicyName*>

*IPaddress* IP address allowed to be the destination of the data (4 decimals separated by periods).

*IPmask* IP network mask (4 decimals separated by periods).

*PolicyName* Name of the IPsec policy to which the destination parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies set dest 192.168.16.0 255.255.255.0 mypolicy`

## IKE IPSEC POLICIES SET DESTPORT

Defines a *destination port* filtering parameter value for the policy. The destination port parameter requires a specific destination port for the data or allows any destination port (*). (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

**ike ipsec policies set destport** <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

*PortNumber*
TELNET
HTTP
SMTP
TFTP
*
: Destination port whose data is allowed by the policy. The port can be specified by one of the listed names or by its number. To allow data through for any destination port, specify an asterisk (*).

*PolicyName*
: Name of the IPsec policy to which the destination port parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Examples:**

```
ike ipsec policies set destport * mypolicy

ike ipsec policies set destport http webpolicy
```

## IKE IPSEC POLICIES SET MODE

Defines the *mode* filtering parameter value for the policy. The mode parameter specifies the encapsulation mode (tunnel or transport) that may be used for the connection (see Transport and Tunnel Encapsulation Modes, on page 119). If no value is set for the mode parameter, tunnel mode is assumed.

**ike ipsec policies set mode** <TUNNEL | TRANSPORT> <*PolicyName*>

TUNNEL
TRANSPORT
: Encapsulation method required for the connection. The default is TUNNEL.

*PolicyName*
: Name of the IPsec policy to which the encapsulation mode parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:**

```
ike ipsec policies set mode transport rtr2rtrpolicy
```

## IKE IPSEC POLICIES SET PEER

Defines a *peer* filtering parameter value for the policy. The peer parameter specifies an IKE peer that may be used for the connection. (The peer must have been defined by IKE peer commands; see IKE Peer Commands, on page 124.)

**ike ipsec policies set peer** <*PeerName*> <*PolicyName*>

*PeerName*
: Name of an IKE peer. To see the IKE peer names, use the **ike peers list** command.

*PolicyName*
: Name of the IPsec policy to which the peer parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:**

```
ike ipsec policies set peer my_aggressive_peer mypolicy
```

## IKE IPSEC POLICIES SET PROPOSAL

Defines a *proposal* filtering parameter value for the policy. The proposal parameter specifies an IKE IPSec proposal that may be used for the connection. (It must have been defined by IKE IPSec proposal commands; see IKE IPSec Proposal Commands, on page 125.)

Unlike the other filtering parameters, the policy may allow more than one value for the *proposal* parameter. For example, two **set proposal** commands could specify two proposals, either of which could be used by the connection. See IKE IPSec Policy Commands, on page 127.

**ike ipsec policies set proposal** <*ProposalName*> <*PolicyName*>

*ProposalName* Name of an IKE proposal. To see the IKE proposal names, use the **ike proposals list** command.

*PolicyName* Name of the IPsec policy to which the proposal parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies set proposal myproposal mypolicy`

## IKE IPSEC POLICIES SET PROTOCOL

Defines a *protocol* filtering parameter value for the policy. The protocol parameter requires a specific protocol that must be used or allows any protocol (*).

**ike ipsec policies set protocol** <*ProtocolNumber* | TCP | UDP | *> <*PolicyName*>

*ProtocolNumber* Protocol required by the policy. The protocol can be specified by number or by name (TCP or UDP). To allow data through for any protocol, specify an asterisk (*).
TCP
UDP
*

*PolicyName* Name of the IPsec policy to which the protocol parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies set protocol * mypolicy`

`ike ipsec policies set protocol tcp webpolicy`

## IKE IPSEC POLICIES SET SOURCE

Defines a *source* filtering parameter value for the policy. The source parameter requires the data come from the specified source IP address and mask. The source is the device or network that sent the packet, not the router that routes the packet.

**ike ipsec policies set source** <*IPaddress*> <*IPmask*> <*PolicyName*>

*IPaddress* IP address allowed to be the source of the data (4 decimals separated by periods).

*IPmask* IP network mask (4 decimals separated by periods).

*PolicyName* Name of the IPsec policy to which the source parameter value is added. To see the policy names, use the **ike ipsec policies list** command.

**Example:** `ike ipsec policies set source 192.168.16.0 255.255.255.0 mypolicy`

## IKE IPSEC POLICIES SET SOURCEPORT

Defines a *source port* filtering parameter value for the policy. The source port parameter requires a specific source port for the data or allows any source port (*) (Because port numbers are TCP and UDP specific, a port filter is effective only when the protocol filter is TCP or UDP.)

**ike ipsec policies set sourceport** <*PortNumber* | TELNET | HTTP | SMTP | TFTP | *> <*PolicyName*>

| | |
|---|---|
| *PortNumber*<br>TELNET<br>HTTP<br>SMTP<br>TFTP<br>* | Source port whose data is allowed by the policy. The port can be specified by one of the listed names or by its number. To allow data through for any source port, specify an asterisk (*). |
| *PolicyName* | Name of the IPsec policy to which the source port parameter value is added. To see the policy names, use the **ike ipsec policies list** command. |

**Example:** `ike ipsec policies set sourceport * mypolicy`

`ike ipsec policies set sourceport http webpolicy`

## IKE IPSEC PROPOSALS ADD

Defines the name of an IKE IPSec proposal. The proposal commands define the proposals exchanged to set up an IPSec security association (SA), that is, an SA to be used for the user data transfer. See IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals add** <*ProposalName*>

| | |
|---|---|
| *ProposalName* | New name for an IPsec proposal. To see the proposal names in use, use the **ike ipsec proposals list** command. |

**Example:** `ike ipsec proposals add myproposal`

## IKE IPSEC PROPOSALS DELETE

Deletes an existing IKE IPSec proposal. For more information, see IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals delete** <*ProposalName*>

| | |
|---|---|
| *ProposalName* | Name of the IPsec proposal to be deleted. To see the proposal names in use, use the **ike ipsec proposals list** command. |

**Example:** `ike ipsec proposals delete myproposal`

## IKE IPSEC PROPOSALS LIST

Lists the IKE IPSec proposals. For more information, see IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals list**

**Example:**

```
# ike ipsec proposals list
IKE IPSEC PROPOSALS:
myproposal
  ESP encryption: 3DES
  ESP authentication: SHA1
  IPComp: None
  Lifetime 600
  Lifedata 50000
```

## IKE IPSEC PROPOSALS SET AHAUTH

Sets the proposal parameter that determines whether AH message authentication is requested and, if it is requested, the hash algorithm used.

**Note:** The proposal must select either the AH or ESP encapsulation methods. It cannot request AH authentication if it requests ESP encryption and/or ESP authentication.

For more information, see ESP and AH Security Protocols, on page 120 or IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set ahauth** <MD5 | SHA1 | NONE> <*ProposalName*>

One of the following:

MD5 — Use AH encapsulation and authenticate using hash algorithm Message Digest 5.

SHA1 — Use AH encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE — No AH encapsulation and no AH message authentication. (If you select this option, ESP encapsulation must be requested by a **set espenc** or **set espauth** command.)

*ProposalName* — Name of the IPsec proposal to which the AH authentication parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set ahauth sha1 myproposal`

## IKE IPSEC PROPOSALS SET ESPAUTH

Sets the proposal parameter that determines whether ESP message authentication is requested and, if it is requested, the hash algorithm used.

For more information, see ESP and AH Security Protocols, on page 120 or IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set espauth** <MD5 | SHA1 | NONE> <*ProposalName*>

One of the following:

MD5 — Use ESP encapsulation and authenticate using hash algorithm Message Digest 5.

SHA1 — Use ESP encapsulation and authenticate using hash algorithm Secure Hash Algorithm-1.

NONE — No ESP encapsulation and no ESP message authentication. (If you select this option, the encapsulation method must be requested by a **set espenc** or **set ahauth** command.)

*ProposalName* Name of the IPsec proposal to which the ESP authentication parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set espauth sha1 myproposal`

## IKE IPSEC PROPOSALS SET ESPENC

Sets the proposal parameter that determines whether ESP encryption is requested and, if it is requested, the encryption method used.

For more information, see ESP and AH Security Protocols, on page 120 or IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set espenc** <DES | 3DES | NULL | NONE> <*ProposalName*>

One of the following:

DES Use ESP encapsulation and 56-bit encryption

3DES Use ESP encapsulation and 168-bit encryption (if 3DES is enabled in the router; see Software Option Keys, on page 99.)

NULL No encryption, but use ESP encapsulation. Headers are inserted as though the data was encrypted. This allows verification of the source, but sends the data in the clear, increasing throughput.

NONE No encryption and no ESP encapsulation. (If you select this option, the encapsulation method must be requested by a **set espauth** or **set ahauth** command.)

*ProposalName* Name of the IPsec proposal to which the ESP encryption parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set espenc 3des myproposal`

## IKE IPSEC PROPOSALS SET IPCOMP

Sets the proposal parameter that requests either no compression or LZS compression. For more information, see IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set ipcomp** <NONE | LZS> <*ProposalName*>

One of the following:

NONE No compression.

LZS Compress using the LZS algorithm.

*ProposalName* Name of the IPsec proposal to which the IP compression parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set ipcomp none myproposal`

## IKE IPSEC PROPOSALS SET LIFEDATA

Sets the proposal parameter that specifies the maximum number of kilobytes for the IPSec SA; 0 means unlimited. After the maximum data is transferred, IKE renegotiates the connection. By limiting the amount of data that can be transferred, you reduce the likelihood of the key being broken.

For more information on proposal parameters, see IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set lifedata** <*kbytes*> <*ProposalName*>

*kbytes* Maximum number of kilobytes transferred before renegotiation; 0 means unlimited.

*ProposalName* Name of the IPsec proposal to which the lifedata parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set lifedata 50000 myproposal`

## IKE IPSEC PROPOSALS SET LIFETIME

Sets the proposal parameter that specifies the length of time (in seconds) before the IPSec SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection.

For more information on proposal parameters, see IKE IPSec Proposal Commands, on page 125.

**ike ipsec proposals set lifetime** <*seconds*> <*ProposalName*>

*seconds* Maximum number of seconds before renegotiation; 0 means unlimited.

*ProposalName* Name of the IPsec proposal to which the lifetime parameter is added. To see the proposal names in use, use the **ike ipsec proposals list** command.

**Example:** `ike ipsec proposals set lifetime 600 myproposal`

## IKE PEERS ADD

Defines the name of a new IKE peer. Other commands specify the address, secret, and mode of the peer connection; see IKE Peer Commands, on page 124.

**ike peers add** <*PeerName*>

*PeerName* New name for an IKE peer. To see the peer names in use, use the **ike peers list** command.

**Example:** `ike peers add my_aggressive_peer`

## IKE PEERS DELETE

Deletes an existing IKE peer entry. For more information, see IKE Peer Commands, on page 124.

**ike peers delete** <*PeerName*>

*PeerName* Name of the IKE peer to be deleted. To see the peer names in use, use the **ike peers list** command.

**Example:** `ike peers delete my_aggressive_peer`

## IKE PEERS LIST

Lists the defined IKE peers. For more information, see IKE Peer Commands, on page 124.

**ike peers list**

**Example:**

```
# ike peers list
IKE Peers:
IKE Peers:
my_aggressive_peer
  IP address = 0.0.0.0
  preshared secret = "confidential_hushhush"
  aggressive, peer id = example.flowpoint.com (Domain name)
             local id = test.flowpoint.com (Domain name)
my_main_peer
  IP address = 1.2.3.4
  preshared secret = "Shipsailsatmidnight"
  main mode
```

## IKE PEERS SET ADDRESS

Sets the IP address of the other endpoint of the secure IKE peer connection. The address specified depends on the mode of the peer connection, which can be either main mode or aggressive mode. (See IKE Management, on page 121.)

If the mode is main mode, the other endpoint of the peer connection is constant, and you specify its IP address.

If the mode is aggressive mode, one end of the connection, the gateway, has a fixed IP address. The other end, the client, has a changing address. When configuring the client, set the peer IP address to the fixed gateway address. When configuring the gateway for an aggressive mode connection, set the peer IP address to **0.0.0.0**.

**ike peers set address** <*IPaddress*> <*PeerName*>

*IPaddress* IP address (4 decimals separated by periods). When configuring a gateway for an aggressive mode connection, set the IP address to **0.0.0.0**.

*PeerName* Name of the IKE peer whose address is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set address 0.0.0.0 my_aggressive_peer`

## IKE PEERS SET LOCALID

Sets the local ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The local ID must match the peer ID on the other end of the connection. The local ID can be an IP address, domain name, or e-mail address as specified by the **set localidtype** command. For more information, see IKE Peer Commands, on page 124.

**ike peers set localid** <*AggressiveModeID*> <*PeerName*>

*AggressiveModeID* IP address (4 decimals separated by periods), domain name, or e-mail address.

*PeerName* Name of the IKE peer whose local ID is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set localid test.flowpoint.com my_aggressive_peer`

## IKE PEERS SET LOCALIDTYPE

Sets the type of the local ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The local ID type must match the peer ID type on the other end of the connection. The possible ID types are IP address, domain name, or e-mail address. For more information, see IKE Peer Commands, on page 124.

**ike peers set localidtype** <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

One of the following:

IPADDR The local ID must be an IP address.
DOMAINNAME The local ID must be a domain name.
EMAIL The local ID must be an e-mail address.

*PeerName* Name of the IKE peer whose local ID type is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set localidtype domainname my_aggressive_peer`

## IKE PEERS SET MODE

Sets the IKE peer connection mode to either main mode or aggressive mode. Main mode is used when the IP addresses of both ends are known and constant. Aggressive mode is used when the address of one end can change, as with a typical modem or DSL connection. (See IKE Management, on page 121.)

**ike peers set mode** <MAIN | AGGRESSIVE> <*PeerName*>

One of the following:

MAIN Selects main mode (both ends constant).
AGGRESSIVE Selects aggressive mode (one end can change).

*PeerName* Name of the IKE peer whose mode is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set mode aggressive my_aggressive_peer`

## IKE PEERS SET PEERID

Sets the peer ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The peer ID must match the local ID on the other end of the connection. The peer ID can be an IP address, domain name, or e-mail address as specified by the **set peeridtype** command. For more information, see IKE Peer Commands, on page 124.

**ike peers set peerid** <*AggressiveModeID*> <*PeerName*>

*AggressiveModeID* IP address (4 decimals separated by periods), domain name, or e-mail address.

*PeerName* Name of the IKE peer whose peer ID is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set peerid example.flowpoint.com my_aggressive_peer`

## IKE PEERS SET PEERIDTYPE

Sets the type of the peer ID for the IKE peer connection. This command is used only when aggressive mode has been selected by the **ike peers set mode** command for this peer name.

The peer ID type must match the local ID type on the other end of the connection. The possible ID types are IP address, domain name, or e-mail address. For more information, see IKE Peer Commands, on page 124.

**ike peers set peeridtype** <IPADDR | DOMAINNAME | EMAIL> <*PeerName*>

One of the following:

IPADDR The peer ID must be an IP address.
DOMAINNAME The peer ID must be a domain name.
EMAIL The peer ID must be an e-mail address.

*PeerName* Name of the IKE peer whose peer ID type is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set peeridtype domainname my_aggressive_peer`

## IKE PEERS SET SECRET

Sets the shared secret for the IKE peer connection. The secret must be identical for both ends. For more information, see IKE Peer Commands, on page 124.

**ike peers set secret** <*secret*> <*PeerName*>

*secret* Secret (up to 256 characters; do not use spaces or non-printable characters).

*PeerName* Name of the IKE peer whose secret is specified. To see the peer names, use the **ike peers list** command.

**Example:** `ike peers set secret confidential_hushhush my_aggressive_peer`

## IKE PROPOSALS ADD

Defines the name of a new IKE proposal. The IKE proposal commands define the proposals exchanged during the Phase 1 SA. For more information, see IKE Management, on page 121.

**ike proposals add** <*ProposalName*>

*ProposalName* Name for the new IKE proposal. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals add my_ike_proposal`

## IKE PROPOSALS DELETE

Deletes an existing IKE proposal. See IKE Proposal Commands, on page 125.

**ike proposals delete** <*ProposalName*>

*ProposalName* Name of the IKE proposal to be deleted. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals delete my_ike_proposal`

## IKE PROPOSALS LIST

Lists the IKE proposals. See IKE Proposal Commands, on page 125.

**ike proposals list**

**Example:**

```
# ike proposals list
IKE proposals:
my_ike_proposal
  Session authentication: Preshared key
  Encryption: DES
  Message authentication: MD5
  DH Group 2
  Lifetime 86400
  Lifedata 0
```

## IKE PROPOSALS SET DH_GROUP

Sets the IKE proposal parameter that specifies the Diffie-Hellman (DH) key generation group used (no group or group 1, 2, or 5). See IKE Proposal Commands, on page 125.

**ike proposals set dh_group** <NONE | 1 | 2 | 5> <*ProposalName*>

One of the following:

| | |
|---|---|
| NONE | No DH group is used. |
| 1 | Use DH group 1. |
| 2 | Use DH group 2. |
| 5 | Use DH group 5. |

*ProposalName* Name of the IKE proposal to which the DH group parameter is added. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals set dh_group 2 my_ike_proposal`

## IKE PROPOSALS SET ENCRYPTION

Sets the IKE proposal parameter that requests *ESP encryption* and specifies the encryption method used. (See IKE Proposal Commands, on page 125.)

**ike proposals set encryption** <DES | 3DES> <*ProposalName*>

One of the following:

DES — Use DES (56-bit) encryption.

3DES — Use 3DES (168-bit) encryption (if 3DES is enabled in the router; see Software Option Keys, on page 99).

*ProposalName* — Name of the IKE proposal to which the encryption parameter is added. To see the proposal names in use, use the **ike proposals list** c ommand.

**Example:** `ike proposals set encryption des my_ike_proposal`

## IKE PROPOSALS SET LIFETIME

Sets the IKE proposal parameter that specifies the length of time (in seconds) before the Phase 1 SA expires; the recommended value is 86400 (24 hours). When the time limit expires, IKE renegotiates the connection. See IKE Management, on page 121.

**ike proposals set lifetime** <*seconds*> <*ProposalName*>

*seconds* — Maximum number of seconds before renegotiation; 0 means unlimited.

*ProposalName* — Name of the IKE proposal to which the lifetime parameter is added. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals set lifetime 86400 my_ike_proposal`

## IKE PROPOSALS SET MESSAGE_AUTH

Sets the IKE proposal parameter that specifies the message authentication done. It can propose no message authentication or authentication using the hash algorithm Message Digest 5 (MD5) or Secure Hash Algorithm-1 (SHA1).

**ike proposals set message_auth** <NONE | MD5 | SHA1> <*ProposalName*>

One of the following:

NONE — No authentication.

MD5 — Authenticate using the MD5 algorithm.

SHA1 — Authenticate using the SHA1 algorithm.

*ProposalName* — Name of the IKE proposal to which the message authentication parameter is added. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals set message_auth md5 my_ike_proposal`

## IKE PROPOSALS SET SESSION_AUTH

Sets the IKE proposal parameter that specifies the session authentication; preshared key is currently the only option. For more information on IKE proposals, see IKE Management, on page 121.

**ike proposals set session_auth** <PRESHARE> <*ProposalName*>

PRESHARE Preshared key.

*ProposalName* Name of the IKE proposal to which the session authentication parameter is added. To see the proposal names in use, use the **ike proposals list** command.

**Example:** `ike proposals set session_auth preshare my_ike_proposal`

# IPSec Commands

The following commands allow you to define an IPSec connection without IKE. To read about IPSec Security, see IPSec (Internet Protocol Security), on page 119.

**Note:** If you define a tunnel using IPSec commands, the keys will remain static. This could pose a security risk and is not recommended. Use of IKE for key management is recommended.

## IPSEC ADD

Defines an IPSec security association (SA) name.

**ipsec add** <*SAname*>

*SAname* Name for the new IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec add show_rx`

## IPSEC DEL

Deletes an existing IPSec security association (SA) name.

**ipsec del** <*SAname*>

*SAname* Name for the IPSec SA to be deleted.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec del show_rx`

## IPSEC DISABLE

Disables a defined IPSec SA entry.

**ipsec disable** <*SAname*>

*SAname* Name for the IPSec SA to be disabled.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec disable show_rx`

## IPSEC ENABLE

Enables a defined IPSec SA entry, indicating it is complete and ready to be used. The command can also re-enable a disabled SA entry.

**ipsec enable** <*SAname*>

*SAname* Name for the IPSec SA to be enabled.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec enable show_rx`

## IPSEC FLUSH

Clears all IPSec definitions.

**ipsec flush**

## IPSEC LIST

Lists one or all IPSec SA entries.

**ipsec list** [<*SAname*>]

*SAname* Optional name for the IPSec SA to be listed.

**Example:**

```
# ipsec list
IPSec security associations:
show_rx
  Gateway: 207.135.89.233
  Inbound
  Tunnel
  BOTH
  3DES
   key=111111111122222222223333333333444444444455555555
  SHA1
   key=aaaaaaaaaabbbbbbbbbbccccccccccddddddddddd (20)
  No compression
  id =424242
  seq=1, bitmap=ffffffff
show_tx
  Gateway: 207.135.89.233
  Outbound
  Tunnel
```

```
  BOTH
  3DES
   key=0123456789012345678901234567890123456789012345678901234567
  SHA1
   key=abcdefabcdefabcdefabcdefabcdefabcdefabcd (20)
No compression
id =123456
seq=6734
```

## IPSEC SET AUTHENTICATION

Selects authentication for the IPSec SA using either SHA-1 (Secure Hashing Algorithm 1) or MD5 (Message Digest 5).

**ipsec set authentication** <MD5 | SHA1> <*SAname*>

One of the following:

| | |
|---|---|
| MD5 | Authenticate using the MD5 algorithm. |
| SHA1 | Authenticate using the SHA1 algorithm. |

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set authentication sha1 show_rx`

## IPSEC SET AUTHKEY

Specifies the authentication key (hexadecimal) for the IPSec SA.

**ipsec set authkey** <*key*> <*SAname*>

*key* Hexadecimal authentication key.

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set authkey aaaaaaaaaabbbbbbbbbbccccccccccddddddddddd show_rx`

## IPSEC SET COMPRESSION

Selects either LZS compression or no compression for the IPSec SA.

**ipsec set compression** <NONE | LZS> <*SAname*>

One of the following:

| | |
|---|---|
| NONE | No compression. |
| LZS | Compress using the LZS algorithm. |

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set compression none show_rx`

## IPSEC SET DIRECTION

Defines the direction of the IPSec SA.

**ipsec set direction** <INBOUND | OUTBOUND> <*SAname*>

One of the following:
INBOUND
OUTBOUND

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set direction inbound show_rx`

## IPSEC SET ENCKEY

Specifies the encryption key.

**ipsec set enckey** <*key*> <*SAname*>

*key* Hexadecimal encryption key (64 bits for DES or 192 bits for 3DES).

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set enckey 1111111111222222222233333333334444444444555555555 show_rx`

## IPSEC SET ENCRYPTION

Selects the encryption used for the IPSec SA: no encryption, DES (56-bit) encryption, or 3DES (168-bit) encryption.

**ipsec set encryption** <NULL | DES-CBC | 3DES> <*SAname*>

One of the following:

NULL Use no encryption.

DES-CBC Use DES (56-bit) encryption.

3DES Use 3DES (168-bit) encryption.

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set encryption 3des show_rx`

## IPSEC SET GATEWAY

Defines the IP address of the gateway of the IPSec SA.

**ipsec set gateway** <*IPaddress*> <*SAname*>

*IPaddress* IP address (4 decimals separated by periods).

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set gateway 207.135.89.233 show_rx`

## IPSEC SET IDENT

Specifies the identifier (SPID) for the IPSec tunnel. It must match the SPID at the other end of the tunnel, that is, the tx SPID on this end must match the rx SPID on the other end.

**ipsec set ident** <*ident*> <*SAname*>

*ident* SPID for the IPSec tunnel.

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set ident 424242 show_rx`

## IPSEC SET MODE

Selects the encapsulation mode (tunnel or transport) for the SA. The default is tunnel mode.

**ipsec set mode** <TUNNEL | TRANSPORT> <*SAname*>

One of the following:
- TUNNEL Tunnel encapsulation mode.
- TRANSPORT Transport encapsulation mode.

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set mode transport rtr2rtr`

## IPSEC SET SERVICE

Selects the authentication and/or encryption services used for the SA.

**ipsec set service** <ESP | AH | BOTH> <*SAname*>

One of the following:
- AH AH authentication.
- ESP ESP encryption.
- BOTH Both ESP encryption and authentication.

*SAname* Name of the IPSec SA.To see the IPSec SA names in use, use the **ipsec list** command.

**Example:** `ipsec set service both show_rx`

# Chapter 6. Managing the Router

This chapter describes the options available for booting software, tells you how to upgrade the router with new releases of software, and explains the process for maintaining copies of configuration files.

## Simple Network Management Protocol (SNMP)

SNMP, a member of the TCP/IP protocol suite, was designed to provide network management interoperability among different vendors' management applications and equipment. SNMP provides for the exchange of messages between a management client and a management agent. The messages contain requests to get or set variables that exist in network nodes, thus allowing a management client to obtain statistics, set configuration parameters and monitor events. These variables (or objects) are defined in Management Information Bases (MIBs), some of which are general or standard SNMP-defined bases. Other bases, such as Enterprise Specific MIBs are defined by different vendors for specific hardware.

The router provides SNMP agent support and support for standard as well as Enterprise Specific MIBs. SNMP is also used internally for configuration of the router. The active SNMP agent within the router accepts SNMP requests for status, statistics, and configuration updates. Communication with the SNMP agent occurs over the LAN or WAN connection.

The supported MIBs and a description of their contents are listed in the following table:

| | |
|---|---|
| **MIB II** | Internet-standard MIB contains only essential elements such as system, interface, addressing, protocol (e.g., IP) and SNMP objects |
| **Bridge MIB** | State/statistics (including spanning tree states) within bridging system |
| **Ethernet MIB** | State/statistics of Ethernet port (e.g., collisions) |
| **IP Forwarding MIB** | State of routing tables (updates MIB II) |
| **PPP MIB For LCP** | State/statistics for each PPP link |
| **Enterprise MIB for configuration** | Router-specific objects for configuration purposes |

Any management application using SNMP over UDP/IP has access to the local SNMP agent. SNMP network management tools vary but often have features to display network maps of SNMP nodes, poll nodes at intervals, trigger alarms on thresholds, graph or list node statistic counters, view and edit individual MIB variables, and print reports.

An example of useful information that can be obtained from a remote SNMP client would be the current status of the router's WAN link and Ethernet interfaces, including protocol (PPP, CSMA-CD), line speed, maximum frame (transmission unit) size, physical address, operating status, or packet traffic rates.

# Telnet Remote Access

The router supports Telnet access. Telnet allows you to log in to the router as if you are directly connected through the Console port. You can issue commands, using the command line interface, to configure the router and perform status monitoring from any remote location. You can use one of the available TCP/IP packages containing the Telnet application. To access the router using Telnet, issue the appropriate command syntax and assign the IP address of the router. You are then directly connected to the router and can issue commands. When you wish to end the Telnet session, exit the application by entering **logoff** or another appropriate command.

A system security timer will log a user off a Telnet session after 10 minutes of inactivity. For more information, refer to the **system securitytimer** command.

Use the command **system telnetport** to enable or disable Telnet access.

# Client TFTP Facility

A client Trivial File Transfer Protocol (TFTP) facility is built into the router that is capable of reading from and writing to the network. A TFTP server must be properly configured to communicate with the router for file transfers to be successful. The client TFTP facility is employed to boot software from a TFTP server, perform software upgrades and copy configuration files to a TFTP server. A TFTP server is integrated into the Windows' Configuration Manager and can also be used as a stand-alone application.

# TFTP Server

The TFTPD (Trivial File Transfer Protocol Daemon) program is installed on your PC as part of the DSL Tools software. TFTPD waits for incoming TFTP requests from TFTP clients. It will put a file on your computer's hard disk or get one from it.

There is no security built into TFTPD, therefore it is important to specify a root directory where all the files that can be accessed are located. When a file is requested, it must be at or below the level of this root directory on your directory tree or the request will be denied. If a TFTP client wants to put a file on your PC, then the file must already exist for writing.

The **Options** menu of the TFTPD program allows the user to configure additional parameters, such as the number of retries and the time between retries. The root directory can also be specified from the **Options** menu.

The DOS command line usage for TFTPD is:

**TFTPD rootdirectory**

The TFTPD operational parameters are kept in the file ROUTER.INI in the form:

rootdir=rootdirectory

retries=maxtries

timeout=timeout

TFTPD is automatically called by BOOTP and Configuration Manager.

# BootP Server

BootP is the Bootstrap Protocol server; it is installed on your PC with the DSL Tools software.

The BootP Server waits for incoming BootP broadcasts from BootP clients. The server looks up the MAC addresses of the incoming BootP request in its database. If the MAC Address is found, the server normally responds to the requestor with an IP address, the IP address of a TFTP server, and the name of a file to use for booting.

# Boot Code

The router provides a number of maintenance options for booting router software. You can boot from the router's FLASH memory, the most common option. Or, you can boot across the LAN network from a TFTP server, perhaps to test a new level of router software before downloading it to FLASH memory. You can also boot through a gateway to a WAN. The router allows you to set permanent network boot parameters used during network booting, and it enables you to temporarily override those parameters. Finally, the router lets you define the order in which the router boot procedures are performed. You can make changes to the boot procedures and specify network boot parameters by entering manual boot mode.

This section provides Boot Mode information for models with configuration switches and models with a reset button.

## Manual Boot Menu

This information applies to most routers with configuration (DIP) switches.

**Note:** For routers with a reset button, see *Recovering Kernels for Routers with a Reset Button, on page 278*.

When the router is shipped, it is set for automatic boot from FLASH memory. If you wish to change the boot options to allow for network booting, change the order of boot procedures, or perform a manual boot, you must enter *manual boot mode*. Automatic and manual boot are controlled by Configuration Switches (on the back panel of the router). The **Options** menu will be displayed if the router's kernel is missing.

### Access Manual Boot Mode

1. Set switch 6 down for Manual Boot mode.
2. Reboot the router by issuing the **reboot** command or by powering up the router.

The router then displays this menu of options:

```
1. Retry start-up
2. Boot from Flash memory
3. Boot from network
4. Boot from specific file
5. Configure boot system
6. Set date and time
7. Set console baud rate
8. Start extended diagnostics
```

**To return to automatic boot mode**

1. When you are ready to return to automatic boot mode, set switch 6 up**.**

2. Reboot by selecting options **1**, **2**, **3,** or **4**. If you reboot with switch 6 in the up position, the router will boot router software automatically in the order and manner that you have specified.

## Option 1: Retry Start-Up

If you are in Manual Boot mode, you can reboot the router in the boot procedure order by selecting option **1**, "Retry start-up"**.** The boot procedure order is either the one you have specified or the default order. The default order is to boot from FLASH memory and then from the network (if defined). If you wish to boot from the network and/or alter the boot procedure order, refer to Option 3: Boot from Network, on page 270.

## Option 2: Boot from FLASH Memory

If you wish to perform a manual boot from FLASH memory, select option **2** from the main boot procedure menu. The router will attempt to boot from FLASH memory. If the boot is unsuccessful, the router will return to manual boot mode. (When you first receive the router, it will default to booting from FLASH during power-up or automatic reboot.)

## Option 3: Boot from Network

First, you need to define permanent network boot parameters using option **5**. Then, select option **3** from the main boot procedure menu to perform a manual boot from the network. The router will attempt to boot from the network using the permanent network boot parameters you have specified.

If you have not defined network boot parameters, the router attempts to locate a BOOTP or RARP server on the network.

BOOTP can be used to supply an IP address, a TFTP Server IP address, and a filename.

RARP is used to obtain an IP address, if it knows the MAC address. The router assumes that the RARP server is also capable of performing the duties of a TFTP server and it will request the filename `KERNEL.F2K` or the filename assigned when permanent network boot parameters are set.)

If a BOOTP or RARP server exists and is properly configured with the router's MAC address, the router will boot from the network. If unsuccessful, the router will return to manual boot mode.

## Option 4: Boot from Specific File

You can temporarily override permanent network boot parameters when you perform a network boot. When the router is in manual boot mode, select option **4**, "boot from specific file", from the main boot procedure menu. Set the network boot parameters; the current default (permanent) parameters are as shown. After you set the parameters, hit the **return** key and the router will boot from the network using the temporary boot parameters. If the boot is unsuccessful, the router will return to manual boot mode.

Once you have installed router software on a network TFTP server, you can have the router boot across the LAN. Network booting requires three parameters:

- the boot IP address
- the TFTP boot server address

- the router software filename on the server

The boot IP address is the router LAN IP address used *during* the boot procedure. This address may differ from the LAN IP address that the router is ultimately assigned. This address is different so that a system can be booted from one subnetwork and then moved to its operational network, if necessary.

The boot IP address is in the form: **zzz.zzz.zzz.zzz**.

The TFTP boot server address is specified as: **xxx.xxx.xxx.xxx** (where xxx.xxx.xxx.xxx is the LAN IP address of the boot server).

The filename must be in the format: **yyyyyyyy.yyy** (similar to the DOS filename format).

Note that once you have set a TFTP server address, it will be assigned to the router software TFTP facility. This server address will then be used whenever a server address is not explicitly specified, including when the **copy** command is in the form:

```
copy tftp:filename kernel.f2k
```

## Option 5: Configure Boot System

a. If you wish to specify permanent network boot parameters, boot the router in manual boot mode.
b. Then select option **5**, "Configure boot system", from the main boot procedure menu to set permanent values.
c. Select options **2**[1], **3**, and **4** to set the three boot parameters described above. After you set permanent network boot parameters, you can change the boot procedure order and/or perform a manual boot from the network.
d. Select option **4** to "Boot through the IP gateway"; in this procedure, the router on the local LAN can boot from a boot server that is not connected directly. Instead, the path to the boot server can include other networks (including WAN, if adequate routers exist). The gateway must be located on the local LAN and reachable by the local router.

You can specify whether the router boots from FLASH first, a network TFTP server first, or never automatically reboots.

1. To set the order, select step **A** under *Configure Boot System*, option **5**.

2. To boot from FLASH first, enter option **1**; to boot from the network first, enter option **2**. If you enter option **3**, the router will always go into manual boot mode; i.e., you must select the boot procedure to be performed.

## Option 6: Set Time and Date

To set the current time and date, boot the router in manual boot mode, and select option **6** from the main boot procedure menu. Set the new date in the format mm[/dd[/yy (or yyyy)]]. Set the new time in military format hh[:mm[:ss]]). You are shown the current date and time. If you set the date to 0/0/0, the real-time clock will be disabled.

**Note:** Your router is Y2K compliant. If you choose to enter only 2 digits to specify the year, values greater than 93 translate to 19xx. Values less or equal to 93 translate to 20xx. The router has a one-hundred-year date range (from 1994 to 2093).

---

1. To reset any parameter, press **enter** following the prompt.

If the date is set to zero, the real-time clock is disabled for long-term storage.

The time and date fields are overwritten by the GUI, when the router is configured by a PC. The time and date values are then read from the PC.

## Option 7: Set Console Baud Rate

Select option **7** to alter the baud rate that is used by the router to communicate over the Console port with the terminal-emulation program. You can override the default rate of 9600. Remember to set the identical baud rate in your terminal emulation program.

## Option 8: Start Extended Diagnostics

Manual boot mode allows you to run extended diagnostics. You may want to run extended diagnostics if you suspect a hardware problem. If you select option **8** from the main boot-procedure menu, you will see the following display:

```
[1] DRAM test
[2] Parity test
[3] POST firmware CRC test
[4] Real-Time Clock chip test
[5] Timers and Interrupts test
[6] Multi-port UART (internal loopback) test
[7] Multi-port HDLC (internal loopback) test
[8] SCC2 External Loopback test
[9] SCC3 External Loopback test
[a] SCC4 External Loopback test
[b] Ethernet Transceiver (internal loopback) test
[-] Deselect all tests
[+] Select all tests
[.] Run selected tests
[#] Enter debugger
[/] Exit extended diagnostics (reboot)
```

Enter the number of each test that you would like to run, or select all tests. Then enter "." to begin diagnostic testing. (All of the tests automatically run when you power up or reboot the router.) A debugging mode is available for use primarily when you encounter a serious problem, in consultation with customer support services.

Boot diagnostics are only available on routers with the MC68EN360 processor.

## Identifying Fatal Boot Failures

Fatal boot failures can be identified by the LEDs light patterns displayed on the front panel of the router.

**Note:** Normal LED states are described in the *Hardware Reference* section of the *Quick Start Guide*.

TEST, LNK, WAN, and LANT display these fatal errors according to the following LED patterns:

| | |
|---|---|
| 0-0-0-G | CPM fail |
| 0-0-G-0 | Timer fail |
| 0-0-G-G | Bad FCS |
| 0-G-0-0 | DRAM fail |
| 0-G-0-G | Interrupt fail |
| 0-G-G-0 | SCC fail |
| Y-0-0-0 | CPU step fail |
| Y-0-0-G | Ethernet loop fail |
| FG-0-0-* | Wait stuck in the boot menu; kernel file could be missing (green LED blinking very rapidly) |
| G-0-0-* | Green occasionally blinks off (at 10-second intervals). The router is issuing BootP requests. |

where

0 = the LED light is off

G = the LED light is on, blinking green

FG = the LED blinking fast

Y = the LED blinking yellow

* = the LED could be on, off, or blinking

Any other combinations of the four LEDs flashing in a regular pattern will indicate an internal error. Should this occur, return the router to the factory for repair or replacement.

**Note:** Non-fatal errors are not displayed by the LEDs, but they do prompt the system to print explanatory messages on the console.

# Software Kernel Upgrades

## Booting and Upgrading from the LAN

You can download a new version of the router software kernel using a TFTP server that already exists on the LAN. The following steps demonstrate how to boot the router software from the network and copy the image from

the network into the router's FLASH memory. When it first connects to the router, the GUI backs up all the files to a directory called Sxxxxx, where x is the router's serial number.

**Note:** We strongly recommend that you use the Configuration Manager's **Upgrade/Backup** tool to upgrade or back up the kernel. The Configuration Manager's tool is more convenient to use than the Command Line Interface.

## Upgrade Instructions

*Read the following steps very carefully before you perform an upgrade:*

1. **Warning:** Before performing this procedure, make sure that you can successfully boot from the network using the manual boot procedure option 3 or 4. Refer to the section Option 3: Boot from Network, on page 270.

2. Copy the router software file KERNEL.F2K (or KERNEL.FPL for an IDSL router) to a directory where it can be accessed by a TFTP server. The TFTP server must be on the same LAN as the target router; i.e., there must not be a router or gateway between the target system and the TFTP server. If the TFTP sever is not on the same network as the target router, enter the gateway in the boot menu as described in the previous section.

3. Log into the Command Line Interface.

4. Enter **reboot** using the Command Line Interface to synchronize the file system and reboot the router. Since the kernel is no longer stored in FLASH memory, the router will try to boot from the network. If you have never set permanent boot parameters, the router attempts to locate a BOOTP or RARP server. If the router successfully reboots from the server, go to step 7.

5. Select option **4** to boot router software from the TFTP server using temporary network boot parameters. You will be prompted for: the router's boot LAN IP address, the TFTP server's IP address, the load address, and the filename of the router's kernel saved on the server. Note that the LAN IP address is the proper address to use during the network boot and this may differ from the IP address ultimately assigned to the router. Enter the temporary network boot parameters (hit the **return** key for the load address). If all entered information is valid, the router will boot from the network. An example follows:

```
Enter selection: 4
  Enter my IP address:
128.1.210.65
  Enter server IP address:
128.1.210.70
  Enter load address [80100]:
  Enter file name: kernel.f2k
```

   Alternatively, select option **5** to set permanent network boot parameters and then boot from the network using option **3**. You would use this option if you wish to boot from the network for a period of time before copying the software to FLASH memory.

6. After the boot is complete, verify that the kernel is running successfully.

7. When you are satisfied that the new kernel is performing as expected, copy the kernel into FLASH memory in the router by typing the two following commands:

**copy tftp@xxx.xxx.xxx.xxx:sfilename kernel.f2k**
**sync**

where xxx.xxx.xxx.xxx is the TFTP server IP address, SFILENAME is the server filename of the kernel, and KERNEL.F2K is the name of the file loaded from FLASH memory by the boot procedure. If you do not specify the server address, a permanent or more recent override TFTP server address will be used, if you have previously defined one. Enter the **sync** command to commit the changes to FLASH memory.

**Warning:** After the kernel is copied, *do not* power down the router until you have issued either a **sync** or **reboot** command to reboot the router. Otherwise the file will not be written to FLASH memory.

8. After successfully copying the kernel to the router, set Configuration switch 2 or 6 to the up position (if you have set it down)**,** and reboot the router from FLASH memory via the **reboot** command. If you have altered the boot procedure order in any way, reset to boot from FLASH memory first. Verify the software revision number by issuing the **vers** command.

   The system is now ready to be re-configured, if necessary. The configuration files are unchanged by the upgrade process.

## Upgrading from the WAN Line

You can download a new version of the router software kernel by using a TFTP server over the WAN line. The following steps show you how to copy the software across the WAN line into the router's FLASH memory.

**Warning:** Before performing this procedure, make sure that you can successfully access the software across the WAN line via a TFTP server.

1. Copy router software KERNEL.F2K to a directory where it can be accessed by a TFTP server.
2. Log in to the Command Line Interface.
3. Copy the kernel into FLASH memory in the router typing the following commands:

   **copy tftp@xxx.xxx.xxx.xxx:sfilename kernel.f2k**
   **sync**

   where xxx.xxx.xxx.xxx is the TFTP server IP address, **sfilename** is the server filename of the kernel, and KERNEL.F2K is the name of the file. If you do not specify the server address, a permanent or more recent override TFTP server address will be used, if you have previously defined one.

**Warning:** After the kernel is copied, *do not* power down the router until you have either issued a **sync** command or rebooted the router. Otherwise the file is not written to FLASH memory**.**

4. After successfully copying the kernel to the router, reboot the router from FLASH memory via the **reboot** command. If a problem occurs during the upgrading process, try the command again (do not reboot until you have successfully copied the kernel). If you have altered the boot procedure order in any way, be sure to reset the router system to boot from FLASH memory first. Verify the software revision number by issuing the **vers** command.

   The router system is now ready to be re-configured if necessary. The configuration files are unchanged by the upgrade process.

# Backup and Restore Configuration Files

To successfully save configuration files to the server, those files must already exist and be writeable by everyone. This restriction is part of the TFTP protocol. Moreover, all the files accessed by the TFTP server must be under a single root directory. Multiple sub-directories can exist below this root directory, but they must be created manually at the server. Neither the sub-directories nor the files can be created remotely.

**Note:** Remember to start the TFTP server from the **DSL Tools** menu.

The **copy** command lets you upload configuration files to the TFTP server where the destination is in the form:

**tftp@xxx.xxx.xxx.xxx:filename.ext**

## Backup Configuration Files (Recommended Procedure)

1. Create a directory under the TFTP root directory corresponding to the system name you want to back up.
2. Create files called SYSTEM.CNF, DHCP.DAT, and FILTER.DAT in this subdirectory. The files can be empty or not, but should be writeable by everyone.

   To back up a copy of the configuration files, enter

   **copy system.cnf tftp@xxx.xxx.xxx.xxx:myname/system.cnf**

   **copy filter.dat tftp@xxx.xxx.xxx.xxx:myname/filter.dat**

   **copy dhcp.dat tftp@xxx.xxx.xxx.xxx:myname/dhcp.dat**

   where **xxx.xxx.xxx.xxx** is the IP address of the TFTP server and **myname** is the router name.

   **Note:** SYSTEM.CNF, FILTER.DAT, and DHCP.DAT are three key files that should be backed up. To see other files that you may also want to save, type the command **dir**.

## Restore Configuration Files

To restore the configuration files, enter:

**copy tftp@xxx.xxx.xxx.xxx:myname/system.cnf system.cnf**

**copy tftp@xxx.xxx.xxx.xxx:myname/filter.dat filter.dat**

**copy tftp@xxx.xxx.xxx.xxx:myname/dhcp.dat dhcp.dat**

**sync**

# FLASH Memory Recovery Procedures

## Recovering Kernels for Routers with Configuration Switches

In the unlikely event that the FLASH file system should become corrupted, there is a series of steps that you can take to attempt to recover. Perform the following procedures in the order listed:

1. Try to repair the file system by issuing the **msfs** command. While logged in, issue a **sync** command followed by an **msfs** command. If the display shows that the file system is corrupted, verify that no other console (via Telnet) is currently modifying the file system with the **ps** command. Then attempt to repair the file system typing the following commands:

   **msfs fix**
   **sync**

2. If the file system is still corrupted (i.e., you cannot write a file) you will have to reformat the file system. First, attempt to save your configuration files as explained in the section *Backup and Restore Configuration Files, on page 276*. Then, while logged in, enter the following commands:

   **format disk**
   **save**
   **copy tftp@xxx.xxx.xxx.xxx:kernel.f2k kernel.f2k**
   **sync**

   The above command sequence assumes that the software presently running from RAM is correctly configured and is still functional. The **save** command re-creates all the configuration files (except the FILTER.DAT file, which you may re-create manually by typing **save filter**). The **copy** command reinstalls the operational software on the FLASH file system, and **sync** commits all this information to disk.

3. In the event that the software running from RAM is not sufficiently functional to perform those steps, you will have to boot from the network using a TFTP server, as explained in the section *Software Kernel Upgrades, on page 273*.

   If you cannot issue the **format** command (as explained in the previous step), you will have to erase the FLASH file system from the boot code.

   a. Flip configuration switch 6 to the down position and reboot the router (by powering down and up again, for example).
   b. At the manual boot menu, select **5**. “Configure boot system”, and enter the “magical” number 98. Then, move switch 6 back to its up position.
   c. Reboot from the network following the steps described in the Software Upgrade Procedure. You will notice error messages indicating that the file system is not formatted. Then log in and enter:

      **format disk**

   d. Recreate the configuration files either by re-entering the information or by restoring them from a TFTP server. Re-install the operational software by entering the command:

      **copy tftp@xxx.xxx.xxx.xxx:kernel.f2k kernel.f2k**

      This assumes that TCP/IP routing is enabled and that an IP address has been assigned to the Ethernet interface.

# Recovering Kernels for Routers with a Reset Button

A router that fails to boot  may be an indication that the kernel has been corrupted.

The following recovery steps can help, but you need to have a kernel for your particular router model. If you installed the DSL Tools and successfully connected to the router, an automatic backup process was started that saved a copy of the kernel and other files to the PC in a subdirectory under DSL Tools called Sxxxxxx, where xxxxxx is the serial number of the unit. The file needed for this recovery is called KERNEL.F2K.

Before you proceed with the recovery steps described below, make sure that the router has a good Ethernet connection to the PC. If a console cable is available, you may want to connect it and start a terminal emulator session so you can see the router's console messages.

Additionally, you can check the LEDs' blinking patterns (on the front panel of the router), which help you identify the state of the router.

## Recovery Steps Using BootP

You may want to connect a console cable, if one is available, and start a terminal emulator session so that you can see the router's console messages.

1. Make sure that the PC path and directory information to a valid kernel are correct.
2. Start the Configuration Manager or Quick Start application (refer to your *Quick Start Guide*).
3. Select Tools and BootP.
4. In the BootP dialog box, enter the following information:
   - The path to the kernel file
   - The serial number of the router
   - The IP address to be used for the boot

   **Note:** This IP address needs to belong to the same subnet as your PC and not be used by another device. For a simple configuration, the IP address 192.168.254.254 will work if your PC already received an IP address from the router when it was still functioning.
5. In the **BootP Setting** dialog box, click **OK**. Configuration Manager writes the above settings to a file called BOOTDBASE.TXT and calls the Bootp server.
6. Power off the router.
7. Insert a a small pen or pointed object into the small reset switch (unlabeled hole) on the back panel of the router (to the right of the Ethernet hub connector). With the object still inserted in the reset switch, power up the router. Wait until all the LED lights flash (about 10 seconds).
8. Once this is accomplished, the BootP server should receive a BootP request and start the TFTPD server. The TFTPD server will send the kernel to the router.
9. Restart Configuration Manager and try to connect to the router. Using the following instructions, you will attempt to write a new kernel to the flash system.
10. From the Configuration Manager's Main Menu, select **Tools** and **Upgrade/Backup**.
11. Click **Firmware** and the **Upgrade** button.

12. Select a kernel file and click **OK**.

Wait until the file is copied, and click **Yes** to reboot the rooter.

# Recovering Passwords and IP Addresses

## Routers with Configuration Switches

**Recover a password:** Set switches 5 and 6 in the down position after the router has booted. With this step, the system password is overridden, thus allowing a forgotten password to be re-entered**.**

**Recover an IP address:** Connect to the console terminal and type the eth list command to find out what the router's IP address is.

## Routers with a Reset Button

The following step will assist you in recovering the router's administrative password or IP address, should you forget them.

Push the reset button and hold it for 3 seconds while the router is running. With this step, the following features are enabled for a period of 10 minutes:

- The system password can be overridden by using the router's serial number as a password.
- A DHCP client address is enabled or created, so that a connected PC can obtain an IP address from the router.

# Batch File Command Execution

This feature is used to load batch files of configuration commands into the router. This allows the user to customize and simplify installation of the router. A script file can contain commands, comments (lines introduced by the # or ; characters), and blank lines.

There are two kinds of script files:

- A one-time script that is executed on startup (only once).
- A group of commands that can be executed at any time from the Command Line Interface with the **execute <***filename***>** command.

One-time scripts are useful to execute the complete configuration process from a default (unconfigured) state.

The following steps describe how to proceed in order to create and execute a one-time script from the Quick Start application.

- Create the script on your PC using Notepad or other text editor. The command syntax can be found in the Command Line Reference manual or enter **?** on the router command line (assuming you have access to the Command Line with the console or with Telnet).

- Select the **Tools | Execute Script** menu item and choose the script file you just prepared. When you click **OK**, the script file is loaded to the router (under the name AUTOEXEC.BAT) and the router is restarted, thus executing the script.

Alternatively, you can manually transfer the script file from your PC to the router using the following method:

- Start the TFTP server on your PC and set the root directory where the script file is located.
- Use the following command to copy the script file to the router file system:
  **copy tftp@** <*PC_IP_address*>:<*PC_file*> <*router_file*>
- To process the commands in the script file, you can either reboot your router (if the script file was copied under the name AUTOEXEC.BAT onto the router) or use the command **execute <***file***>**.

**Note:** If present, the file AUTOEXEC.BAT is renamed AUTOEXEC.OLD before it is executed, so that it is only run once. If you clear the router configuration with the **Reset Defaults** button of the **Upgrade/Backup** tool or the **reboot default** command, the AUTOEXEC.OLD is renamed back to AUTOEXEC.BAT and re-run after the boot up, thus restoring your configuration.

You can include the commands **rename** <*autoexec.old*> <*autoexec.bat*> or **reboot** in a script file; there is no limitation on the commands that you might define in your scripts. The **rename** command is useful if you need the script to execute on every startup, whereas the **reboot** command is useful to apply changes and have them take effect (almost) immediately.

**Caution:** If you create a one-time script file (copied to the router under the name AUTOEXEC.BAT), do not include the commands **rename** <*autoexec.old*> < *autoexec.bat*> *and* **reboot** at the same time. This will result in an endless loop of starting the router, executing the script, restarting the router, re-executing the script.

# Chapter 7. Troubleshooting

Software problems usually occur when the router's software configuration contains incomplete or incorrect information. This chapter discusses:

- Diagnostic tools that are available to help identify and solve problems that may occur with your router
- Symptoms of software configuration problems
- Actions for you to take
- System messages

## Diagnostic Tools

This section describes three diagnostic tools available to you:

- The LEDs on the front panel of your router.
- The History Log the lists the router's activity.
- The **ping** command that can verify IP connectivity.

### Using LEDs

Most hardware problems can be diagnosed and solved by checking the LEDs on the front panel of your router. The following table summarizes the *normal* LED sequence in the left column (five consecutive states) from **Power On** to **Ready State**. The right column lists suggestions for problems reflected by an "abnormal" LED state (no progression to the next state).

For the LED patterns that indicate fatal boot errors, see Identifying Fatal Boot Failures, on page 273.

Note that the normal LED progression involves:

PWR LED (power LED)

TEST LED (self-test indicator LED)

LINK LED (modem link)

**If the Power (PWR) LED is off:**

- Check that the power cord is firmly plugged into the back panel of the router and the other end into an active AC wall or power-strip outlet.
- Check that the power switch is turned on.

| Normal LED Sequence | State Length | Problem<br>If the LED sequence stops at this stage: |
|---|---|---|
| **State 1**<br>**Power ON**<br>PWR - green<br>TEST - amber<br>LINK - off | 5 sec | A hardware problem has been detected.<br>Contact Technical Support. |
| **State 2**<br>All lights flash | 1 sec | |
| **State 3**<br>PWR - green<br>TEST - green<br>LINK - off | 5 sec | 1. Check that the DIP switches are all up.<br>2. Check that the correct software was loaded. |
| **State 4**<br>PWR - green<br>TEST - green<br>LINK - amber | 5 to 10 sec | 1. Check your DSL cable.<br>2. Check the physical connection from your router to the DSLAM (Central Office).<br>3. Possible problem with DSLAM card. |
| **State 5**<br>PWR - green<br>TEST - green<br>LINK - green | Ready State | |

Once the router is in **Ready State**, the other LEDs may indicate transmitting and receiving activity as follows:

- The WAN LED indicates that the WAN is transmitting activity.
- The LANT LED indicates that the Ethernet LAN is transmitting activity.
- The LANR LED indicates that the Ethernet LAN is receiving activity.

# History Log

The **History Log** utility is a troubleshooting tool which displays the router's activity. It can be accessed from a terminal emulation session (including Configuration Manager) or from Telnet.

To see message explanations, refer to the System Messages section, page 290.

## Accessing History Log through Telnet

1. Click **Connect** and then **Remote System.**
2. Enter the router's IP address.
3. Click **Connect.**

### Accessing History Log through Configuration Manager

1. Select **Tools** and **Terminal Window** (the console cable is required).
2. Log in with your administration password into the router (e.g. **"admin"**).
3. Use the command **system history** to view the buffer contents.

### Other Logging Commands

- If you wish to monitor your router activity at all times, use the command **system log start** to view a continuous log, using Telnet. (This command will not work in a Terminal Window session; it only works from Telnet.)
- The command **system log status** is used to find out if other users, including yourself, are using this utility.
- To discontinue the log at the console, use the command **system log stop.**

When you exit Telnet, you automatically stop any logging programs running in that session.

**Note: History Log** is preserved across reboots, but not across power outages or power down.

## Ping Command

You can verify IP connectivity to the router by running a **ping** command. You will probably find a ping utility bundled with your TCP/IP stack. In Microsoft Windows, the command is called `PING.EXE` and can be found in your Windows directory.

**Note:** Before using the **ping** command to troubleshoot, make sure that the PWR, TEST, and LINK lights are green.

### Instructions for Windows 95

#### 1. Start a DOS window

a. Select **Start** from the Windows 95 taskbar.

b. Select **Programs**.

c. Select **MS-DOS Prompt**.

#### 2. Issue the ping command

In the DOS window, type the command:

**ping <***IP address***>**

**Example:** `ping 192.168.254.254`

### Interpretation and Troubleshooting

To isolate a problem with the TCP/IP protocol, perform the following three tests:

1. Try to **ping** the IP address of your PC. If you get a response, proceed directly with step 2. If you don't get a response, check that:
   - The network adapter card is installed.
   - The TCP/IP protocol is installed.
   - The TCP/IP protocol is bound to the network adapter.
2. Try to **ping** the IP address of your router. If you get a response, proceed directly to step 3. If you don't get a response, the problem lies between your PC and router:
   - Check the cables.
   - Check the hub.
   - Make sure that your PC and the local router are in the same IP subnetwork.
3. Try to **ping** the DNS server. Write down the results and call your Network Service Provider.

# Investigating Hardware Installation Problems

When investigating a hardware installation problem, first **check the LEDs** on the front panel of the router. Many common hardware problems can be easily diagnosed by the LED indicators. For more information, refer to this chapter's section entitled *Diagnostic Tools, Using LEDs,* page 281.

**If the terminal window display has a problem:**

- Ensure your console is plugged in and turned on.
- Verify that you are on the right communications port (Com1, Com2).
- Check the configuration parameters for speed, parity, etc. Make sure the console is not in an XOFF state. Try entering a **`"ctrl q"`**.
- Verify that the RS232 device attached to the console is configured as a DTE. If not, a crossover or null modem adapter is required.

If the **factory configuration** has a problem:

- Compare the router configuration with your router order.
- Verify that the model number is correct (the number is displayed during the boot procedure). The model number and serial number are also displayed on the main window of Configuration Manager.

# Investigating Software Configuration Problems

This section suggests what to do if you cannot:

- connect to the router.
- log in.
- access the remote network.
- access the router via Telnet.
- download software.

Finally, if you have a VoDSL router,  it suggests how to trouble-shoot your telephony services.

## Problems Connecting to the Router

If you cannot connect your PC to the target router for configuration:

- For a LAN connection, verify that the router's IP address matches the IP address previously stored into the router's configuration. You must have previously set the router's Ethernet LAN IP address and subnet mask, saved the Ethernet configuration changes, and rebooted the router for the new IP address to take effect.
- Check that your LAN cable is pinned correctly and each pin end is securely plugged in.
  **Note:** If you are using a straight-through cable, the colors for pins 1, 2, 3, and 6 should match on both connectors. If you are using a crossover cable, the colors for pins 1, 2, 3, and 6 on one connector should match respectively 3, 6, 1, and 2 on the other connector.
- Make sure the PC and target router are on the same IP subnetwork or the target router is reachable through a router on your LAN. They can, however, be on different networks if IP routing is *off*.
- Check Network TCP/IP properties under Windows 95 and the control panel of the TCP/IP driver installed under Windows 3.1.
- Check if the LAN LED on the router's front panel blinks when "pinged".
- Check your Ethernet board IRQ settings: the PC's table may have become "confused". If so, reboot your PC.

## Problems with the Login Password

You have been prompted for the login password and received the following message:  Login Password is invalid.

- Type the correct password and press enter. Remember that the password is case-sensitive. If the password is **admin**, check that you are entering it in lowercase and that the Caps key is not active.
- If you have forgotten the password, you must reset the login password. Refer to the *User Guide*, *Changing Configuration Switches,* and perform the following procedure:

  1. Move switches **5** and **6** down.
  2. Type **login <***newpasswd***>.** Password checking is overridden.
  3. Move switches **5** and **6** up**.**
  4. Complete any configuration update that caused the prompt for login.

5. Change your login password to a new password.
6. Store the configuration and reboot the router.

**Note:** If you do not reset switches **5** and **6** to the up position and then reboot, the router is placed in maintenance mode. Set switches **5** and **6** up and turn the power off and then on again.

## Problems Accessing the Remote Network

## Bridging

- Make sure to reboot if you have made any bridging destination or control changes.
- All IP addresses must be in the same IP subnetwork (IP is being bridged).
- Check that a bridging default destination has been configured and is enabled.
- Be sure to reboot if the bridging destination or status has been changed.
- Check that bridging is enabled locally (use the **remote listBridge** command, page 196).
- Verify that bridging is enabled by the remote router (use the **remote list** command, page 195).
- Verify that the authentication passwords are correct.
- Reboot your PC if you have Windows for WorkGroups.
- In Windows 95, do not forget to declare shared disk directories. Check the sharing properties on your **C: drive .**
- In the Terminal Window, check that calls are answered from the remote router.
- Check also for any PAP/CHAP errors for the remote router.

## TCP/IP Routing

- Check that Ethernet LAN TCP/IP Routing has been enabled (**eth list** command, page 181).
- The IP addresses of the local and remote networks belong to different IP subnetworks.
- Make sure that there is an existing route to the remote network.
- Make sure that there is a route back from the remote network.
- There must be a source WAN IP address defined if you are using NAT.
- Check that, if required, the source and remote WAN IP addresses are on the same subnetwork
- Reboot if you have made any IP address or control or protocol option changes.
- Check that the IP address of the station/network connected to the LAN beyond the remote router is correct, as well as the associated subnet mask.
- If the remote router WAN IP address and subnet mask are required, check that they have been specified correctly.
- Check that a default route has been specified, if needed.
- Be sure to reboot if IP addresses or control or protocol option changes have been made.

- Check that you are using an Ethernet cable.
- Check that IP routing is enabled at both ends.
- The IP address must be within the valid range for the subnet.
- Verify that the IP and gateway addresses are correct on the PC.
- Windows 95 may remember MAC addresses: if you have changed MAC addresses, reboot the router and the PC.
- In Windows 3.1., check that the TCP driver is installed correctly. Ping (**ping** command) your PC's IP address from the PC. Successful "pinging" results let you know that the TCP driver is working properly.
- If you have changed an IP address to map to a different MAC device, and **ping** or IP fails, reboot your PC.
- Use the **iproutes** command (page 139) to verify which router's name is the default gateway (this cannot be 0.0.0.0).

## IPX Routing

- Check that IPX routing has been enabled and that the remote end is enabled for IPX routing.
- Validate that the IPX WAN network number matches the remote router's WAN network number.
- Check that IPX SAPs correctly identify the servers and applications on the remote network and have valid network numbers, node numbers, etc.
- Check that every SAP has a router to its internal network.
- Check that the IPX routes (network numbers, hops, and ticks) seeded into the routing table for network segments and servers beyond the remote router are correct.
- Validate that the IPX WAN network number matches the remote router's WAN network number.
- Check that the IPX routes (network numbers, hops, and ticks) seeded into the routing table for network segments and servers beyond the remote router are correct.
- Check that IPX SAPs correctly identify the servers and applications on the remote network and have valid network numbers, node numbers, etc.
- Be sure to reboot if IPX addresses, routes, SAPs or control has been changed.
- If the router fails to negotiate IPX:
    - Make sure that at least one WAN number is not equal to zero at one end of the link.
    - The server must have an IPX route to the remote LAN.
    - The Novell server needs to have **burst mode** turned on.
    - Large Internet packets have to be turned on.
- For Novell 3.12 and later:
    - Client needs VLM.EXE, net.cfg: large Internet packets=ON, Pburst=5
- If you can't see the server SAPs:

- Check the frame types using the **eth list** command (page 181) and ensure that they are the same on both routers.
- Check that the Ethernet cable is correctly plugged in.
- Make sure that the Novell server is up.

## Incorrect VPI/VCI (ATM Routers)

If you are given an incorrect VCI/VPI number or none at all to use for the remote, and you need to determine what the possible value might be, use the **atom findpvc** command (see ATM Debug Commands, on page 294).

## Problems Accessing the Router via Telnet

- Ensure that the router has a valid IP address.
- Check that the Ethernet cable is plugged in.

## Problems Downloading Software

- Ensure that a TFTP server is properly set up to locate the router software.
- Verify that the router is loading from the network and not from FLASH memory.

## Trouble-Shooting Telephony Services

After the router WAN link activates (the LINK LED is green), you should get a dial tone. The dial tone should be received even if you have not yet configured your IP and bridge network settings.

If you do not get a dial tone, check the following:

- Does the router have power?
- Is the local phone cord plugged in?
- Is the voice PVC set correctly in the router? (See the following debug commands.)
- Is the WAN link down? (The LINK LED should be solid green.)
- Is the DSLAM provisioned for the second PVC?
- Is the voice gateway connected and provisioned?
- Is the ATM network down between the DSLAM and the voice gateway?

If you hear clicking during heavy data downloads, check that the DSLAM supports quality of service (QOS) and that the ATM switch has the voice PVC provisioned for vRT and the data at a lower priority.

The Port Monitor GUI program can show you the voice PVC and the last event message. The following debug commands may also be helpful.

**ifs** Shows whether the data and voice PVC's are configured and percent loading.

**For ATM routers:**

**atom voice** Displays the voice PVC.

**atom voice** <*x***y*> Changes the voice PVC to the specified *x***y*.

**For Frame Relay routers:**

**frame voice** Displays the voice DLCI.

**frame voice** <*x*> Changes the voice DLCI to the specified number *x*.

**frame stats** Shows LMI statistics.

For a **frame stats** example, see page 150.

If the voice gateway is a Jetstream gateway, the following commands are available:

**voice l2stats** Shows AAL2 statistics for control messages.

**voice l2clear** Clears the AAL2 statistics to 0.

To see the CRC and line errors for SDSL, enter:

**sdsl stats**

For an **sdsl stats** example, see page 225.

# System Messages

System messages are displayed on the terminal and sent to a log file (if you have opened one). The messages listed in this section are time-stamped informational and error messages. The messages are in the following format:

**dd+hh:mm:ss:nn sysfunc: message**

| | |
|---|---|
| *dd* | date in xx/xx/xx format as specified during router initialization |
| *hh:mm:ss:nn* | time in military format (hours:minutes:seconds:hundredths of seconds) |
| *sysfunc* | software function |
| *message* | message |

The following are examples of messages:

```
12/05/1997-16:31:17:ADSL: Startup initiated
12/05/1997-16:36:26:ADSL: Startup handshake in progress
```

## Time-Stamped Messages

<router/user> didn't negotiate our IP address correctly
Explanation: The remote router did not negotiate the IP address options as was expected by the local router.

<router/user> terminated IPCP prematurely
Explanation: IP failed to negotiate. Try to change the remote or the source WAN IP address.

Far Avg SQ #: <2-digit number> dB [4-digit number]
Explanation: Message about the average signal quality for the remote router. This information appears during modem startup and should be ignored unless requested by Technical Support.

Authorization failed
Explanation: PAP cannot be negotiated.

Can't agree with <router/user> on what their IP address should be
Explanation: The IP address entry for the remote router in the remote router database does not match what the local router expects.

Can't obtain an IP address from <router/user>: one is needed in single user mode
Informative message.

Can't supply an IP address to <router/user>
Explanation: The remote end requests an IP address from the local end, which cannot supply it.

Cannot remove SYSTEM.CNF
Informative message.

Connecting to <router/user> @ <number> over <link/number>
Explanation: The local router is trying to connect to the specified remote destination.

Data Mode
Explanation: The connection is established and operational.

Duplicate IPX route to <router/user>
Explanation: There exist two routes to the same IPX destination. Remove one of the routes.

Duplicate IPX SAP <SAP number> to <router/user>
Explanation: There exist two IPX SAPs for the same IPX destination. Remove one of the SAPs.

Duplicate route <IP route> found on remote <router/user>
Explanation: There exist two IP routes to the same IP destination. One route needs to be removed.

Idle
Explanation: Data is not being transmitted.

IP is configured for numbered mode with <router/user>, but no address for it
Explanation: On one end of the connection, remote entries have been configured for numbered mode. On the other end, remote entries have been configured for unnumbered mode. Neither end cab communicate with the other.

No Signal Detected -- Check WAN Cable!
Explanation: (SDSL-specific error message) Your SDSL router cannot establish connectivity. Check your physical line.

No system name known - using defaults
Explanation: The router does not have a system name. For PAP/CHAP negotiation, the router will use a default name and password.

Note: IPX is misconfigured for <router/user> - no IPX WAN network
Explanation: IPX WAN address is wrong or missing.

Note: There is no IPX route statically defined for <router/user>
Informational message.

PPP: Peer not negotiating <IP | BNCP | IPX | CCP> right now
Explanation: One end of the network is not negotiating the same protocol as the other end.

Remote <router/user> didn't accept our CHAP password
Informational message.

Remote <router/user> does not respond to LPC echo. Link closed
The connection was terminated.

Remote <router/user> on <channel> didn't authenticate in time
Explanation: PPP authentication protocol did not succeed.

Remote <router/user> refuses to authenticate
Informational message.

Remote <router/user> tried to use PAP when CHAP was expected
Explanation: The remote end negotiated PAP while its minimum security level in the remote database was set to CHAP.

Remote <router/user> used wrong password <CHAP | PAP>
Explanation: The remote end has used an invalid password during CHAP or PAP security authentication.

Remote didn't accept our CHAP password
Explanation: The router attempted CHAP security authentication but the remote end rejected the password.

Remote on <interface> didn't authenticate in time
Informational message.

Remote on <interface> rejected our password with PAP
Informational message.

Remote on <interface> refuses to authenticate with us
Explanation: The remote destination refused to participate in the PAP/CHAP authentication process.

Startup failed
Explanation: The ATM modem could not synchronize with the remote end. Call Technical Support.

Startup failed: failure code = <number>, Status [code]
Explanation: The ATM modem could not synchronize with the remote end. Call Technical Support

TelnetD
Explanation: Connection accepted. A remote configuration session has been established.

User <router/user> is disabled in remote database
Informative message.

User <router/user> not found in remote database <PAP | CHAP>
Explanation: The authentication is coming from an unknown remote router.

# How to Obtain Technical Support

**Before you contact Technical Support, please have the following information ready:**

- Router model number
- Router software version
- Date of purchase
- Type of operating system (Windows 95, 98, NT, or Windows for Workgroups)
- Description of the problem
- List of other equipment such as personal computers, modems, etc. and third-party software you are using, including revision levels.

To determine how to contact Technical Support, see the Quick Start Guide that came with your router, or contact:

**Efficient Networks, Inc. West**

**Mail:** **5200 Great America Parkway**
**Santa Clara, CA 95054**

**Phone:** **+1 (408) 878-6810**

**Fax:** **+1 (408) 878-6801**

**Email:** **support@efficient.com**

**Web:** **http://www.efficient.com**

# Debugging Commands

The following commands may be available for debugging purposes. Please use them with caution because they are not fully supported.

## General Debug Commands

**ifs**

Shows which interfaces are configured or active. For an example of its output, see page 138.

**mlp debug** <LCP | NCP | BNCP | IPCP | IPXCP | CCP | ECP | MLP | AUTH | NCPSTATES> [<0>]

BNCP is for bridging, ECP for encryption, and NCPSTATES for state table changes.

To turn off the trace, enter the command with the optional **0** at the end.

**ipdebug icmp 1**

**ipdebug nat 1**

These commands show data received. The **ipdebug icmp 1** command is useful for showing the router can receive cells ok.

**dod whycall 80**

Prints out the packet that is causing the link to come up. This is useful when **system onewan on** is set. (This command makes PVC's look like dial-up links, that is, the link comes up only if user traffic exists and the link times out on inactivity.) For more information, see SYSTEM ONEWANDIALUP, on page 164.

**dod debug** <1 | 0>

Shows trace of when we bring up the link or time out link on inactivity. Specify 1 to turn on the trace; specify 0 to turn off the trace.

**ping** [-c *count*] [-i *wait*] [-s | -l *size*] [-I *sourceipaddr*] <*ipaddr*>

Sends an echo message to the specified IP address. You cannot ping your own LAN address; you can ping your own WAN address.

You can set the length of user data down to 0 bytes (**-s 0** or **-l 0**) so in routing mode it fits in one ATM cell. For more information, see PING, on page 141.

**system log** [start | stop | status]

Starts event logging when logged in via Telnet. Otherwise, you don't see any event messages. It is not needed if you are using a console cable. For more information, see SYSTEM LOG, on page 162.

**system support**

Dumps all tables. If you capture and send this output to Technical Support, it can be useful in debugging problems. For more information, see SYSTEM SUPPORTTRACE, on page 165.

The information dumped includes the history log and information about the version, memory, processes, the file system, general system information, Ethernet, DHCP, Voice, remote database, interfaces, bridging, the ARP table, IP routes, IPX routes, IPX SAPs, L2TP tunnels, and IP filters.

**copy /RAW-IMAGE ttp@192.4.210.171:test**

Uses the special file name /RAW-IMAGE to copy all of flash memory to a backup file for system debugging.

## ATM Debug Commands

**atom findPVC** <on | off>

Shows VPI*VCI of cells received. This command is normally used to find the ATM VPI*VCI number necessary for configuring a remote when the Service Provider either has supplied the wrong value or simply is not able to supply one.This command should only be used when there are *no* remotes defined or when the remote entries are disabled.

The command output is directed to the console. If Telnet is used to log into the router, then issue the **system log start** command to direct the console output to the Telnet session.

**Example:**

```
# atom findPVC on
No remote entry found with PVC (VPI*VCI) 1*2
```

In this case, an ATM VPI*VCI is found for which there is no remote defined. 1 is the number of the VPI as found in the ATM stream. 2 is the number of the VCI as found in the ATM stream. The discovered number may be used as the VPI*VCI value in the remote, for determining whether communications are possible.

**atom echoPVC** <vpi number>*<vci number>

Enables an echo pvc (use **atom echo 0*21**). This is configured automatically and can be disabled with **atom echo 0*0**. The echoPVC will echo back any ATM cell received on the PVC exactly as received. This is useful when an administrative service wishes to ensure ATM connectivity but cannot use ATM OAM F5 cells to achieve this function.

**atom pls** <on | off>

Changes payload scrambling.

**atom empty** <ATMF | ITU>

Changes type of ATM empty cell sent or expected. It is useful if ATM sync delination errors when combined with **ato stats** command.

**atom dumpunknowncells** [on | off]

Looks at the content of an ATM cell. It will not affect normal operation performance.

## Web GUI Debug Commands

Using a browser pointed to the router, you can display special pages in the web GUI, such as, 192.18.254.254/time.htm.

**time.htm** Displays time.

**quick.htm** Displays all Easy Setup parameters.

**factory.htm** Resets all values to factory defaults.

**dump.htm** Shows all values.

## SDSL Debug Commands

**sdsl ***

Displays all available SDSL commands.

**sdsl btstat**

Displays available status values.

Example:

```
# sdsl bts
Available status:
   SLM ........................... Input Signal Level
   DC_METER ...................... Input DC Offset
   FELM .......................... Far-End Signal Attenuation (Cal'd
at 1168 Kbs)
   NMR ........................... Noise Margin
   TIMING_RECOVERY_CONTROL ....... Timing Recovery Control
   STARTUP_STATUS ................ Bit-Pump Status
   BIT_PUMP_PRESENT .............. Bit-Pump Present
   SELF_TEST ..................... Self Test
   REGISTER ...................... Read Register
   CONFIGURATION ................. Big-Pump Configuration
   STAGE_NUMBER .................. Stage Number
   AAGC_VALUE .................... AAGC
   READ_TX ....................... Read Tx Gain
   BER_METER_STATUS .............. BER Meter Status
```

**sdsl btstat ***

Displays available SDSL status commands.

**sdsl bts felm**

Displays Far-End Signal Attenuation. It gives an estimate of the length of the loop.

Output example: `SDSL: FELM: 63 [0x3f]`

**sdsl bts nmr**

Displays noise margin. Large values are symptoms of a bad or excessively lengthy loop.

Output example: `SDSL: NMR: 224 [0xe0]`

**sdsl states trace** [<all>]

Turns on trace of line changes. To turn off the trace, append **all** to the command.

Example:

```
# sdsl states trace
```

```
SDSL State Trace [00000001]: states => s
# sdsl states trace all
SDSL State Trace [00000000]: off
```

**sdsl huh**

Dumps various registers.

Example:

```
# sdsl huh
SDSL:
    Bitpump: 8973
    CPE -- ACTIVATING
    Line Rate: [AUTO] 192 Kb/s  [3072 KHz]
    Activation Interval: 99 [AUTO:20]  [symbol_rate: 24]
    AutoSpeed:
      FastSearchAttemptsPerPass: 2
      FastSearchPasses.........: 2
      SlowSearchAttemptsPerPass: 5
      SaveDelayInSeconds.......: 45
    Two Symbol Time: 23 uS
    FW: V4.3    CS 5: BR = 80000401  OR = fffff8f66
    Ints -- On : 1228462  Mask: 0b00  IRQ: 02
    BP Status Reads: 0
    BT assumed on other end!
    BT - Self Test will run
    SDSL CONFIGURATION: 0x03f9  20 LOST: 10 [0x0a] Sym Rate: 24 [0x18]
```

## Voice Router Debug Commands

**voice 12stats**

Shows AAL2 statistics for voice router.

**voice 12stats clear**

Resets values.

**dsp** <NOEC | ECON>

Turns echo canceller on (NOEC) or off (ECON).

The following commands can be used for standalone phone verification. (This is for lab or bench verification only.)

**ds init nobort** Starts DSP for this test.

**ds cas x** Connects and rings port x.

**ds ploop x-y** Connects port x to y.

**ds init** Reinitialize after testing.

For example, to connect port 1 to port 2, use this command sequence:

```
ds init nobort
```

```
ds cas 1
ds cas 2
ds ploop 1-2
```

## ADSL DMT Router Debug Commands

**dmt ***

Displays the available DMT commands.

**dmt ver**

Displays the code version of line driver.

**dmt speed**

Displays the speed of the link.

**dmt ms**

Shows the modem status.

**dmt link <DEFAULT | T1_413 | G_DMT | G_LITE | MULTIMODE>**

Sets the link type. It is used to force the CPE into ANSI (T1.413), G_DMT, or G_LITE mode. DEFAULT and MULTIMODE are the same. The link type survives reboots.

## Frame Relay Debug Commands

**frame stats**

Displays statistics. For more information, see FRAME STATS, on page 150.

## ATM Tracing Commands

**atom print**

Shows count of good and bad atm cells and frames.

**atom rx** <on | off>

Shows AAL5 frames received.

**atom promisc on**

Turns on promiscuous mode (rx ATM cells no matter what VPI*VCI).

**atom cellrx** <on | off>

Traces ATM cells received.

**atom tx** <on | off>

Traces ATM cells sent.

**atom stats** <*n*>

Prints the ATM statistics every *n* seconds. It shows good and bad cells and frames.

## IP Filtering Debug Commands

The following commands can start and stop an IP filter watch. For more information about IP filters, see IP Filtering, on page 103.

**eth ip filter watch** <on | off>

**remote ipfilter watch** <on | off>

Prints a message to the console if a packet to or from this remote is dropped or rejected.

# Appendix A. Network Information Worksheets

To configure the target (local) router, you need to fill out one of the following blank worksheet(s) that applies to your Link Protocol/Network Protocol situation:

- PPP with IP routing configuration
- PPP with IPX routing configuration
- PPP with bridging configuration
- RFC 1483/RFC 1490 with IP routing configuration
- RFC 1483/RFC 1490 with IPX routing configuration
- RFC 1483/RFC 1490 with bridging configuration
- RFC 1483MER/ RFC 1490MER with IP routing configuration
- FRF8 with IP routing configuration

A blank worksheet is also available to enter information needed to configure a Dual-Ethernet router with IP routing enabled.

### If you are connecting to more than one remote router:

You need to fill out one set of information for each remote router in the *Remote Routers* section of the worksheet.

### If you are setting up both ends of the network:

You will need a *mirror image* of the information listed in your target router worksheet in order to configure the router on the other end of the WAN link.

**Note:** You may want to review the Chapter 4. Sample Configuration on page 100.

# Configuring PPP with IP Routing

| PPP with IP Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your settings** |
| **System Settings** | | |
| System Name | **system name <***name***>** | ..... |
| System Message | **system msg <***message***>** | ..... |
| Authentication Password | **system passwd <***password***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipaddr***> <***ipnetmask***>** [**<***port#>*]| ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver<***ipaddr***>** | ..... |
| Change Login | **system admin <***password***>** | ..... |
| **Remote Routers** | | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol | **remote setProtocol PPP <***remoteName***>** | ..... |
| PVC or DLCI | **remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>**<br>**remote setDLCI <***number***><***remoteName***>** | ..... |
| Security | **remote setAuthen <***protocol***> <***remoteName***>** | ..... |
| Remote's Password | **remote setOurPasswd <***passwd***> <***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <***remoteName***>** | ..... |
| TCP/IP Route Address | **remote addIproute <***ipnet***> <***ipnetmask***> <***hops***>**<br>**<***remoteName***>** | ..... |
| If NAT is enabled:<br>To enable NAT -and-<br>You may need to enter a Source WAN Port Address | **remote setIpTranslate** on **<***remoteName***>**<br>**remote setSrcIpAddr <***ipaddr***> <***ipnetmask***>**<br>**<***remoteName>* | ..... |
| If NAT is OFF:<br>You may need to enter a Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***> <***ipnetmask***>**<br>**<***remoteName***>** | ..... |
| **IP and IPX Routing** | | |
| TCP/IP Routing (Internet Firewall) | **eth ip enable**<br>**eth ip firewall <**on | off**>** | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

# Configuring PPP with IPX Routing

| PPP with IPX Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| **System Settings** | | |
| System Name | **system name <***name***>** | ..... |
| System Message | **system msg <***message***>** | ..... |
| Authentication Passwd | **system passwd <***password***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipnet***> <***ipnetmask***>** [**<***port#***>**] | ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>** | ..... |
| Change Login | **dhcp set valueoption domainnameserver <***ipaddr***>** | ..... |
| Ethernet IPX Network # | **system admin <***password***>**<br>**eth ipx addr <***ipxnet***>** [**<***port#***>**]<br>**eth ipx frame <***type***>** | ..... |
| **Remote Routers** | | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol | **remote setProtocol PPP <***remoteName***>** | ..... |
| PVC or DLCI | **remote setPVC <***vpi number***>*<***vci number***>** **<***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** | ..... |
| Security | **remote setAuthen <***protocol***> <***remoteName***>** | ..... |
| Remote's Password | **remote setPasswd <***password***> <***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <***remoteName***>** | ..... |
| IPX Routes Add | **remote addIpxroute <***ipxNet***> <***metric***> <***ticks***>** **<***remoteName***>** | ..... |
| IPX SAPs Add | **remote addIpxsap <***servicename***> <***ipxNet***>** **<***ipxNode***> <***socket***> <***type***> <***hops***>** **<***remoteName***>**<br>**remote setIpxaddr** *<ipxNet> <remoteName>* | ..... |
| **IP and IPX Routing** | | |
| TCP/IP Routing | **eth ip disable** | ..... |
| IPX Routing | **eth ipx enable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

# Configuring PPP with Bridging

| PPP with Bridging | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| | **System Settings** | |
| System Name | **system name <***name***>** | ..... |
| System Message | **system msg <***message***>** | ..... |
| Authorization Password | **system passwd <***password***>** | ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoptiondomainnameserver <***ipaddr>* | ..... |
| Change Login | **system admin <***password***>** | ..... |
| | **Remote Routers** | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol | **remote setProtocol PPP <***remoteName***>** | ..... |
| PVC or DLCI | **remote setPVC <***vpi* number**>*<***<***vci number***>**<br>**<***remoteName***>**<br>**remote setDLCI <***number***> <***remoteName***>** | ..... |
| Security | **remote setAuthen <***protocol***> <***remoteName***>** | ..... |
| Remote's Password | **remote setOurPasswd <***password***>**<br>**<***remoteName***>** | ..... |
| Bridging On/Off | **remote enaBridge <***remoteName***>** | ..... |
| | **IP and IPX Routing** | |
| IP Routing | **eth ip disable** | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

# Configuring RFC 1483 / RFC 1490 with IP Routing

| RFC 1483 / RFC 1490 with IP Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| | **System Settings** | |
| System Message | **system msg <***message***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipnet***> <***ipnetmask***>** [*port#***>**] | ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>** | ..... |
| | **dhcp set valueoption domainnameserver <** *ipaddr***>** | ..... |
| Change Login | **system admin <** *password***>** | ..... |
| | **Remote Routers** | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol/PVC[a] (for ATM routers) | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <***vpi number***> * <***vci number***>** *<remoteName>* | ..... |
| Link Protocol /DLCI[b] (for Frame Relay Routers) | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <***number***><***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <***remoteName***>** | ..... |
| TCP/IP Route Address | **remote addiproute <***ipnet***> <***ipnetmask***>**<br>**<***hops***> <***remoteName***>** | ..... |
| If NAT is enabled:<br>To enable NAT<br>-and-<br>You must enter a Source WAN Port Address | **remote setIpTranslate on <***remoteName***>**<br>**remote setSrcIpAddr <***ipaddr***> <***ipnetmask***>**<br>**<***remoteName***>** | ..... |
| If NAT is OFF:<br>You may need to enter a Source WAN Port Address | **remote setSrcIpAddr <***ipaddr***> <***ipnetmask***>**<br>*<remoteName>* | ..... |
| | **IP and IPX Routing** | |
| TCP/IP Routing (Internet Firewall) | **eth ip enable**<br>**eth ip firewall** <on \| off> | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

# Configuring RFC 1483 / RFC 1490 with IPX Routing

| RFC 1483 / RFC 1490 with IPX Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| **System Settings** | | |
| System Message | **system msg <***message***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipaddr***> <***ipnetmask***>** [*port#>*] | ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** | ..... |
| Ethernet IPX Network # | **eth ipx addr <***ipxnet***>** [*>port#>*]<br>**eth ipx frame <***type***>** | ..... |
| Change Login | **system admin <***password***>** | ..... |
| **Remote Routers** | | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol/PVC[a] (for ATM routers) | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** | ..... |
| Link Protocol/DLCI[b] (for Frame Relay Routers) | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <***number***><***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <***remoteName***>** | ..... |
| IPX Routes Add | **remote addIpxroute <***ipxNet***> <***metric***> <***ticks***>**<br>**<***remoteName***>** | ..... |
| IPX SAPs Add | **remote addIpxsap <***servicename***> <***ipxNet***>**<br>**<***ipxNode***> <***socket***> <***type***> <***hops***>**<br>**<***remoteName***>**<br>**remote setIpxaddr <***ipxNet***> <***remoteName***>** | ..... |
| **IP and IPX Routing** | | |
| TCP/IP Routing (Internet Firewall) | **eth ip disable**<br>**eth ip firewall** <on \| off > | ..... |
| IPX Routing | **eth ipx enable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

# Configuring RFC 1483 / RFC 1490 with Bridging

| RFC 1483 / RFC 1490 with Bridging | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| **System Settings** | | |
| System Message | **system msg <***message***>** | ..... |
| DHCP Settings | **dhcp set valueoption domainname <***domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** | ..... |
| Change Login | **system admin <***password***>** | ..... |
| **Remote Routers** | | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol/PVC[a] (for ATM routers) | **remote setProtocol RFC1483 <***remoteName***>**<br>**remote setPVC <**vpi *number***>*<***vci number***><br>**<***remoteName***>** | ..... |
| Link Protocol /DLCI[b] (for Frame Relay Routers) | **remote setProtocol FR <***remoteName***>**<br>**remote setDLCI <***number***><***remoteName***>** | ..... |
| Bridging On/Off | **remote enaBridge <***remoteName***>** | ..... |
| **IP and IPX Routing** | | |
| IP Routing | **eth ip disable** | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.

# Configuring RFC 1483MER / RFC 1490MER with IP Routing

| RFC 1483MER/RFC 1490MER with IP Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| | **System Settings** | |
| System Message | **system msg <***message***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipaddr***> <***ipnetmask***>**[<*port#*>] | ..... |
| DHCP Settings | **dhcp set valueoption domainname <** *domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr* **>** | ..... |
| Change Login | **system admin <***password***>** | ..... |
| | **Remote Routers** | |
| New Entry | **remote add <***remoteName***>** | ..... |
| Link Protocol/PVC[a] (for ATM routers) | **remote setProtocol RFC1483MER <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** | ..... |
| Link Protocol /DLCI[b] (for Frame Relay Routers) | **remote setProtocol MER <***remoteName***>**<br>**remote setDLCI <***number***><***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <** *remoteName***>** | ..... |
| TCP/IP Route Address | **remote addIproute <***ipnet***>**<br>**<***ipnetmask***><***ipgateway***> <***hops***>**<br>**<***remoteName***>** | ..... |
| If NAT is enabled:<br>To enable NAT,enter:<br>-and-<br>Enter a Source WAN Port Addr | **remote setIpTranslate on <***remoteName***>**<br>**remote setSrcIpAddr <***ipaddr***> <***ipnetmask***>**<br>**<***remoteName***>** [c] | ..... |
| If NAT is not enabled:<br>You may need to enter a Source WAN Port Addr | **remote setSrcIpAddr** <*ipaddr*> <*ipnetmask*><br><*remoteName*> [d] | ..... |
| | **IP and IPX Routing** | |
| TCP/IP Routing (Internet Firewall) | **eth ip enable**<br>**eth ip firewall** <on \| off> | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

a Enter this information if you are using RFC 1483 in an ATM environment.
b Enter this information if you are using RFC 1490 in a Frame-Relay environment.
c The mask is the mask of the remote network.
d The mask is the mask of the remote network.

# Configuring FRF8 with IP Routing

| RFC 1483FR with IP Routing | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| **System Settings** | | |
| System Message | **system msg <***message***>** | ..... |
| Ethernet IP Address | **eth ip addr <***ipaddr***> <***ipnetmask***>** [<*port#*>] | ..... |
| DHCP Settings | **dhcp set valueoption domainname <** *domainname***>**<br>**dhcp set valueoption domainnameserver <***ipaddr***>** | ..... |
| Change Login | **system admin <***password***>** | ..... |
| **Remote Routers** | | |
| New Entry | **remote add** <*remoteName*> | ..... |
| Link Protocol/PVC | **remote setProtocol FRF8 <***remoteName***>**<br>**remote setPVC <***vpi number***>*<***vci number***>**<br>**<***remoteName***>** | ..... |
| Bridging On/Off | **remote disBridge <***remoteName***>** | ..... |
| TCP/IP Route Address | **remoteaddIproute <***ipnet***> <***ipnetmask***>**<br>**<***hops***><***remoteName***>** | ..... |
| If NAT is enabled:<br>To enable NAT<br>-AND-<br>You must enter a Source WAN Port Addr | **remote setIpTranslate on <***remoteName***>**<br>**remote setSrcIpAddr <***ipaddr***> <***mask***>**<br>**<***remoteName***>** [a] | ..... |
| If NAT is not enabled:<br>You may need to enter a Source WAN Port Addr | **remote setSrcIpAddr <***ipaddr***> <***mask***>**<br>**<***remoteName***>** [b] | ..... |
| **IP and IPX Routing** | | |
| TCP/IP Routing<br>(Internet Firewall) | **eth ip enable**<br>**eth ip firewall <**on \| off**>** | ..... |
| IPX Routing | **eth ipx disable** | ..... |
| **Store**<br>**Reboot** | **save**<br>**reboot** | |

a The mask is the mask of the remote network
b The mask is the mask of the remote network

# Configuring a Dual-Ethernet Router for IP Routing

This table outlines commands used to configure a Dual-Ethernet router for IP Routing.

| **Dual-Ethernet Router - IP Routing** | | |
|---|---|---|
| **Steps** | **Commands** | **Your Settings** |
| | **System Settings** | |
| System Name | **system name** <*name*> | ..... |
| Message | **system msg** <*message*> | ..... |
| | **Ethernet Settings** | |
| Routing/ Bridging Controls | **eth ip enable**<br>**eth br disable** | ..... |
| ETH/0 IP Address | **eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>] | ..... |
| ETH/1 IP Address | **eth ip addr** <*ipaddr*> <*ipnetmask*> [<*port#*>] | ..... |
| TCP/IP default route address | **eth ip addroute** <*ipaddr*> <*ipnetmask*> <*gateway*> <*hops*> [<*port#*>] | ..... |
| | **DHCP Settings** | |
| Define DHCP network for ETH/1 | **dhcp add** [<*net*> <*mask*> **|** <*ipaddr*> **|** <*code*> <*min*> <*max*> <*type*> | ..... |
| Create an address pool for ETH/1 | **dhcp set addresses** <*first ipaddr*> <*last ipaddr*> | ..... |
| DNS Domain Name | **dhcp set valueoption domainname <***domainname***>** | ..... |
| DNS Server | **dhcp set valueoption domainnameserver** <*ipaddr*> | ..... |
| WINS Server Address | **dhcp set valueoption winsserver** <*ipaddr*> | ..... |
| **Store** | **save** | |
| **Reboot** | **reboot** | |

# Appendix B. Configuring IPX Routing

## IPX Routing Concepts

To establish IPX Routing, you will need to enter all remote routers in the remote router database to which your router will connect.

1. For each remote router, enter the network addresses and services that may be accessed beyond the remote router.
2. Also enter a network number for the WAN link.
3. After you have specified the route addressing and services, you can then enable IPX routing across the Ethernet LAN.

**Static Seeding:** When IPX traffic is destined for network segments and servers beyond the remote router, the target router's routing information table must be statically seeded. Static seeding ensures that the target router connects to the appropriate remote router. After the link is established, RIP broadcast packets will dynamically add to the target router's routing table. Seeding the routing table is not necessary for target routers that never connect; they will discover remote networks beyond the calling router as soon as RIP updates arrive (provided the remote router supports RIP). However, for watchdog spoofing to work, you will need to define the remote IPX routes for network segments and servers.

## Configure IPX Routing

Configuring your router for IPX routing can be rather complex. The following section will guide you through the configuration process. Remember that PPP authentication configuration must be completed *before* you attempt IPX routing configuration. The full router configuration for simple IPX routing includes the following:

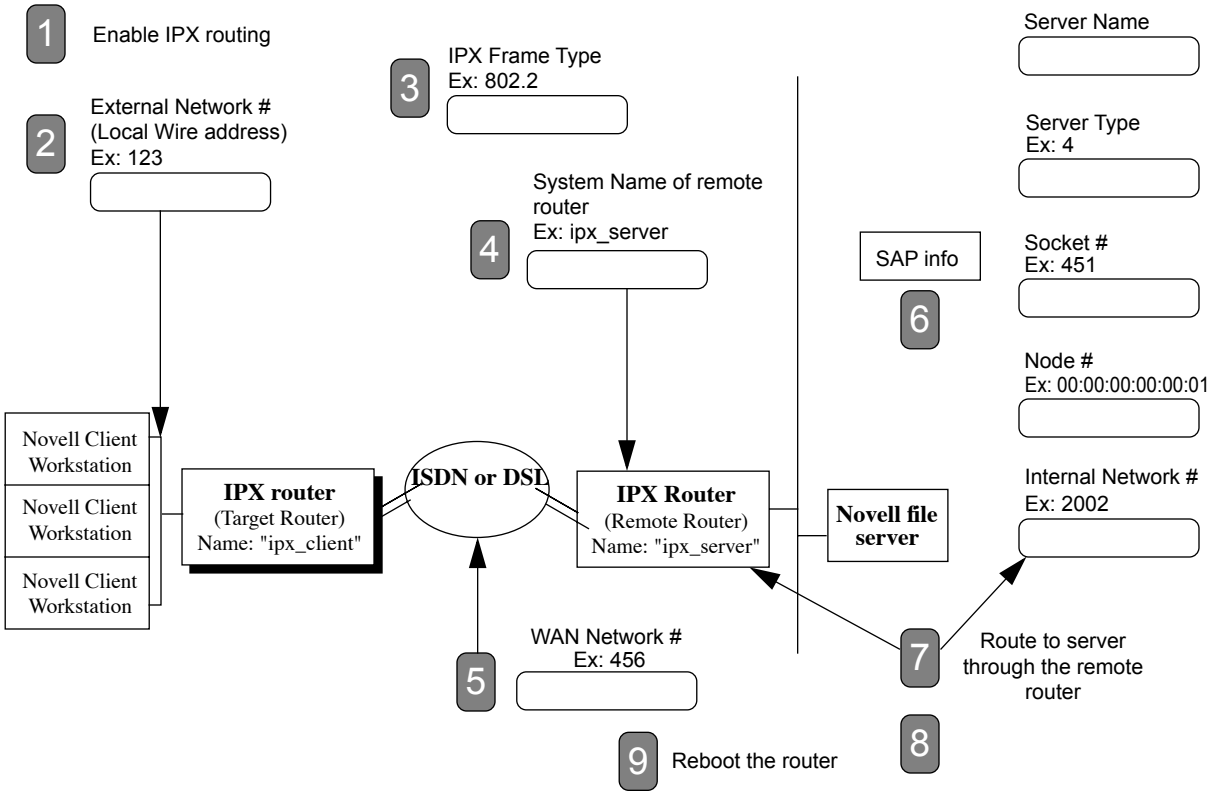
- PPP authentication
- IPX routing (this section)

The following section, Step 1: Collect Your Network Information for the Target (Local) Router, on page 310, provides a configuration diagram and a command table to assist you with the configuration of the target router.

Step 2: Review your Settings, on page 311 lists the commands used to review the IPX configuration and provides a configuration example.

# Step 1: Collect Your Network Information for the Target (Local) Router

**The remote side of the WAN link has all of the file and print services.**
**Enter the needed network information in the blank boxes of the diagram. Then match the boxes' numbers with the numbers in the Command Table below to configure the target router for IPX.**




## Command Table

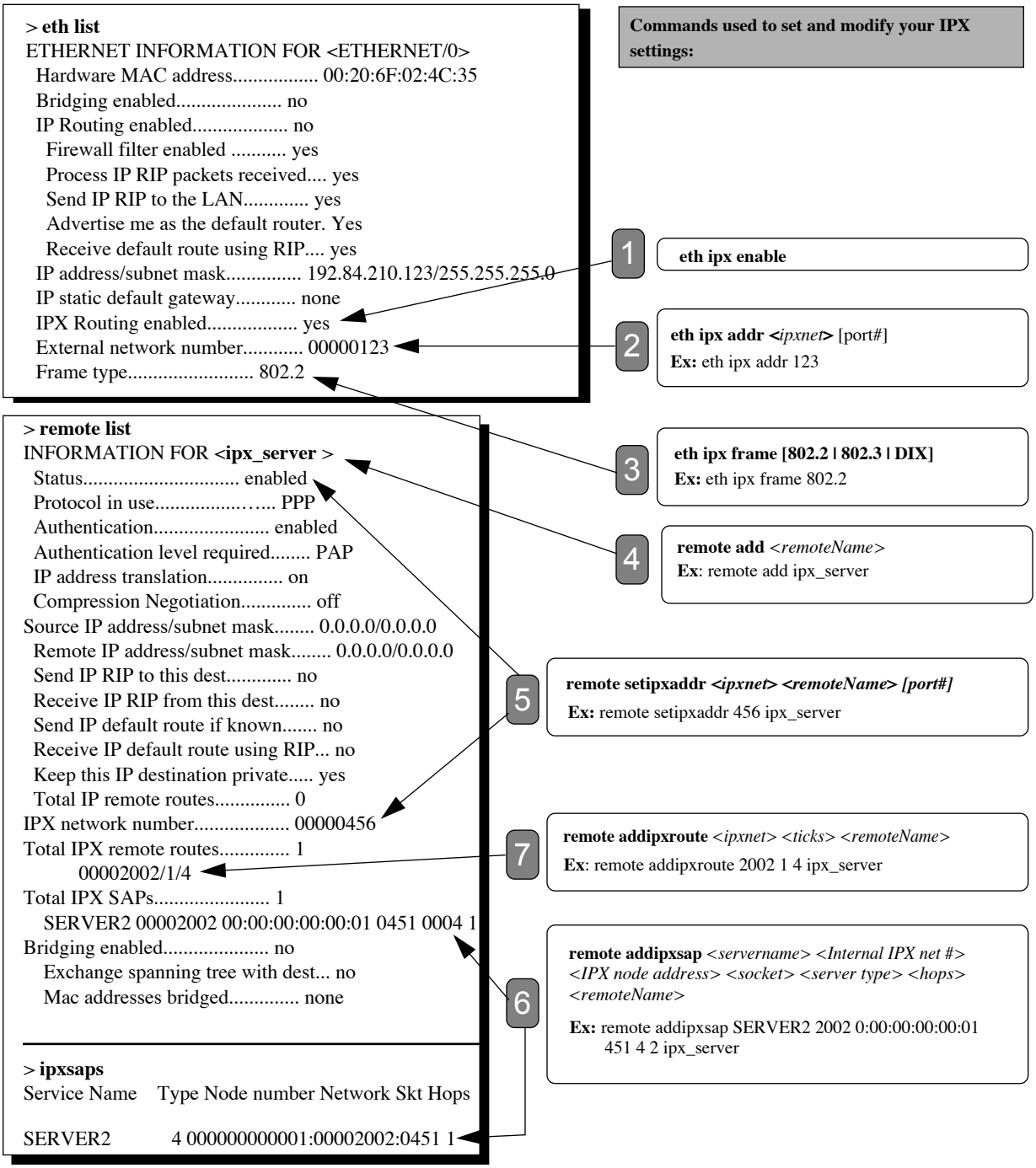
These commands are used to configure the Target (client-side) router (**ipx_client**). Log in with the password **admin.**

| IPX Commands with examples | Ref # | Comments |
|---|---|---|
| **eth ipx enable** | **1** | Enable IPX Routing |
| **eth ipx addr** 123 | **2** | Set the local 'wire' address |
| **eth ipx frame** 802.2 | **3** | Set the Frame Type |
| **remote add** ipx_server | **4** | Add a connection name |
| **remote setIpxaddr** 456 ipx_server | **5** | Set the WAN network # (common to both sides) |
| **remote addIpxsap** SERVER2 2002 00:00:00:00:00:01 0451 4 1 ipx_server | **6** | Add a file server (SAP) |
| **remote addIpxroute** 2002 1 4 ipx_server | **7** | Add a route to the server |
| **save** | **8** | Save your settings |
| **reboot** | **9** | Reboot for changes to take effect |

# Step 2: Review your Settings

**Commands used to review your IPX configuration:**

- **eth list**
- **remote list**
- **ipxsaps**



# Appendix C. Accessing the Command Line Interface

This section provides step-by-step instructions on how to connect the PC to the Console Port of the router. It then describes how to access the Command Line Interface from different environments.

# Connect the PC to the Console Port of the Router

For local access, the PC (or ASCII) terminal is connected to the **Console** port of the router.

To connect your PC to the console port of the router, you will need:

- One TPE straight-through cable (red-labeled cable provided - if already in use, you need to obtain another straight-through cable)
- One RJ-45 to DB-9 adapter (provided)

1. Connect the PC to the console port of the router as follows:
2. Plug one of the TPE straight-through cable's RJ-45 connectors into the console port (the rightmost port on the back panel of the router).
3. Attach the RJ-45 to DB-9 adapter to the cable's other connector and connect it to your workstation's serial port (usually Com 1 or Com 2).

# Access the Command Line Interface

Once the router is powered on and the PC is connected to the console port, you are ready to access the Command Line Interface. It can be accessed from:

- The terminal window from within the Configuration Manager application (for local access)
- The terminal window from within the Quick Start application
- A terminal session under Windows (for local access)
- A terminal session under a non-Windows platform (for local access)
- A Telnet session (for remote access)

## Terminal Window under Configuration Manager

To access the terminal window from within the Configuration Manager application, click **Tools** and **Terminal Window** from the main menu. The menu selection **Commands** provides shortcuts to most of the commands described in this manual. These shortcuts will substantially reduce your amount of keying.

## Terminal Window under Quick Start

To access the terminal window from within the Quick Start application, click **Tools** and **Terminal Window** from the main menu. The menu selection **Commands** provides shortcuts to most of the commands described in this manual. These shortcuts will substantially reduce your amount of keying.

## Terminal Session under Windows (HyperTerminal)

1. To open the HyperTerminal emulator available in Windows, click **Start** on your desktop, select **Programs, Accessories**, and **HyperTerminal**.
2. Double-click **Hypertrm.exe.**
3. Enter a name for the connection in **Connection Description**.
4. In the **Phone Number** window, under **Connect using**, select **Choose Direct to Com 1** (or **2**).
5. In the **Com 1** (or **2**) **Properties page**, enter the following port settings:
    - 9600 baud
    - 8 data
    - No parity
    - 1 stop bit

## Terminal Session for a Non-Windows Platform (Macintosh or UNIX)

To open a terminal window emulation in a Macintosh or UNIX environment, you will need a VT100 terminal emulation program. Refer to your system documentation to determine which communications programs to use to communicate with the router's console interface.

- Start your VT100 terminal emulator and configure it with the following port settings:
- 9600 baud
- 8 data
- No parity
- 1 stop bit
- Appropriate Com port

## Telnet Session

1. Make sure that your PC and router's addresses are in the same subnetwork. For example, the address of the router is 192.168.254.254 and the address of the PC is 192.168.254.253.

2. Click the **Start** button on your PC desktop, click **Programs**, **Windows Explorer**, **Windows** and **TELNET.EXE**.

3. Select **Connect** and **Remote System** from the menu.

4. In the **Connect** window, enter the router's IP address. Click **Connect**.

# Index

## Numerics

3DES encryption, 126, 254

## A

accessing the Command Line Interface, 313
address translation, 91
ADPCM voice encoding, 18
ADSL DMT router debug commands, 297
AH IPSec protocol, 119, 120
ATM, 210
ATM debug commands, 294
ATM tracing commands, 297
atom dumpunknowncells command, 294
atom echoPVC command, 294
atom empty command, 294
atom pls command, 294
atom voice command, 288
authentication
    ESP message, 126, 253
    ESP protocol, 120
    IKE, 122
    IKE message, 125
    IKE session, 125
Authentication Header protocol, 119

## B

Backing up configuration files, 276
binding a virtual route
    for a remote interface, 186
    for an Ethernet interface, 169
boot code, 269
    manual boot mode (configuration switches models), 269
    manual boot mode (reset button models), 278
boot failures, 273
boot options
    baud rate for console, 272
    booting from the network, 270
    extended diagnostics, 272
    manual boot mode, 269
    time and date, 271
BootP server, 269
bridge filtering, 78
bridge filtering commands, 246
bridging
    configuration information (for dual-Ethernet router), 42
    configuration information (with PPP), 35
    configuration information (with RFC 1483), 38
    configuration information (with RFC 1490), 38
    configuration table (with PPP), 48, 49
    configuration table (with RFC 1483/RFC 1490), 52
    general information, 16
    test, 57

## C

clearing the configuration, 144
Command Line Interface
    access, 313
Command Line Interface conventions, 135
commands
    ?, 136
    adsl ?, 208
    adsl restart, 208
    adsl speed, 208
    adsl stats, 209
    arp delete, 136
    arp list, 137
    atm ?, 210
    atm pcr, 210
    atm save, 210
    atm speed, 211
    atom dumpunknowncells, 294
    atom echoPVC, 294
    atom empty, 294
    atom findPVC, 294
    atom pls, 294
    atom voice, 288
    bi, 137
    bi list, 137
    call, 138
    copy, 146

debugging, 293
delete, 147
dhcp ?, 227
dhcp add, 227
dhcp bootp allow, 228
dhcp bootp disallow, 228
dhcp bootp file, 228
dhcp bootp tftpserver, 229
dhcp clear addresses, 229
dhcp clear expire, 229
dhcp clear valueoption, 230
dhcp del, 230
dhcp disable, 230
dhcp enable, 231
dhcp list, 231
dhcp list definedoptions, 232
dhcp list lease, 234
dhcp relay, 234
dhcp set addresses, 235
dhcp set expire, 235
dhcp set lease, 235
dhcp set mask, 237
dhcp set otherserver, 236
dhcp set valueoption, 237
dir, 147
dmt, 297
dmt mode, 212
dod, 293
erase filter, 145
eth ?, 167
eth add, 167
eth br disable, 213
eth br enable, 213
eth delete, 168
eth ip addHostMapping, 214
eth ip addr, 168
eth ip addroute, 169
eth ip addServer, 214
eth ip bindroute, 169
eth ip delHostMapping, 215
eth ip delServer, 215
eth ip directedbcast, 171
eth ip disable, 172
eth ip enable, 172
eth ip filter, 172
eth ip firewall, 176
eth ip mgmt, 176
eth ip options, 178
eth ip translate, 216
eth ipx addr, 179
eth ipx disable, 180
eth ipx enable, 180
eth ipx frame, 180
eth list, 181
execute, 147
exit, 138
filter br ?, 246
filter br add, 246
filter br del, 246
filter br list, 247
filter br use, 247
format disk, 148
frame, 149
frame stats, 138
frame voice, 289
hdsl ?, 217
hdsl save, 218
hdsl speed, 218
hdsl terminal, 218
help, 136
ifs, 138
ike flush, 248
ike ipsec policies add, 248
ike ipsec policies delete, 248
ike ipsec policies disable, 248
ike ipsec policies enable, 249
ike ipsec policies list, 249
ike ipsec policies set dest, 249
ike ipsec policies set destport, 250
ike ipsec policies set mode, 250
ike ipsec policies set peer, 250
ike ipsec policies set proposal, 251
ike ipsec policies set protocol, 251
ike ipsec policies set source, 251
ike ipsec policies set sourceport, 252
ike ipsec proposals add, 252
ike ipsec proposals delete, 252
ike ipsec proposals list, 252
ike ipsec proposals set ahauth, 253
ike ipsec proposals set espauth, 253
ike ipsec proposals set espenc, 254
ike ipsec proposals set ipcomp, 254

ike ipsec proposals set lifedata, 254
ike ipsec proposals set lifetime, 255
ike peers add, 255
ike peers delete, 255
ike peers list, 255
ike peers set address, 256
ike peers set localid, 256
ike peers set localidtype, 257
ike peers set mode, 257
ike peers set peerid, 257
ike peers set peeridtype, 258
ike peers set secret, 258
ike proposals add, 258
ike proposals delete, 259
ike proposals list, 259
ike proposals set dh_group, 259
ike proposals set encryption, 260
ike proposals set lifetime, 260
ike proposals set message_auth, 260
ike proposals set session_auth, 261
ipdebug, 293
ipifs, 139
iproutes, 139
ipsec add, 261
ipsec del, 261
ipsec disable, 261
ipsec enable, 262
ipsec flush, 262
ipsec list, 262
ipsec set authentication, 263
ipsec set authkey, 263
ipsec set compression, 263
ipsec set direction, 264
ipsec set enckey, 264
ipsec set encryption, 264
ipsec set gateway, 264
ipsec set ident, 265
ipsec set mode, 265
ipsec set service, 265
ipxroutes, 139
ipxsaps, 140
isdn ?, 220
isdn list, 220
isdn save, 220
isdn set switch, 221
l2tp ?, 238
l2tp add, 238
l2tp call, 239
l2tp close, 240
l2tp del, 240
l2tp forward, 240
l2tp list, 241
l2tp set address, 238
l2tp set authen, 239
l2tp set chapsecret, 239
l2tp set dialout, 241
l2tp set hiddenAVP, 242
l2tp set ourpassword, 242
l2tp set oursysname, 242
l2tp set ourTunnelName, 242
l2tp set remoteName, 243
l2tp set type, 243
l2tp set window, 244
logout, 140
mem, 140
mlp, 293
mlp summary, 141
msfs, 148
ping, 141
ps, 142
reboot, 143
remote ?, 182
remote add, 183
remote addHostMapping, 183
remote addIpRoute, 183
remote addIpxRoute, 184
remote addIpxSap, 185
remote addServer, 185
remote bindIPVirtualRoute, 186
remote del, 186
remote delencryption, 187
remote delhostmapping, 187
remote delIpRoute, 187
remote delIpxRoute, 188
remote delIpxSap, 188
remote delOurPasswd, 189
remote delOurSysName, 189
remote delserver, 189
remote disable, 190
remote disAuthen, 190
remote disBridge, 190
remote enaAuthen, 190

remote enable, 191
remote enaBridge, 191
remote ipfilter, 191
remote list, 195
remote listBridge, 196
remote listIpRoute, 196
remote listIpxroutes, 197
remote listIpxSaps, 197
remote listPhones, 198
remote setATMtraffic, 211
remote setAuthen, 198
remote setBrOptions, 199
remote setCompression, 199
remote setDLCI, 221
remote setEncryption (Diffie-Hellman), 200
remote setEncryption (PPP DES), 200
remote setIpOptions, 201
remote setIPTranslate, 202
remote setIpxaddr, 202
remote setl2tpclient, 202, 244
remote setLNS, 202
remote setlns, 244
remote setMgmtIpAddr, 203
remote setmgmtipaddr, 203
remote setOurPasswd, 203
remote setOurSysName, 204
remote setPasswd, 204
remote setProtocol, 204
remote setprotocol, 221
remote setPVC, 205
remote setRmtIpAddr, 205
remote setSrcIpAddr, 206
remote stats, 206
remote statsclear, 207
remote unbindIPVirtualRoute, 207
rename, 149
save, 143
save all, 143
save atm25, 144
save atom, 144
save dhcp, 144
save dod, 144
save eth, 144
save filter, 144, 145
save sys, 144, 145
sdsl ?, 223
sdsl btstat, 295
sdsl eoc, 223
sdsl huh, 296
sdsl save, 224
sdsl speed, 224
sdsl states trace, 295
sdsl stats, 225
sdsl terminal, 226
sync, 149
system ?, 153, 156, 161
system addhostmapping, 154
system addhttpfilter, 154
system addiproutingtable, 154
system addserver, 155
system addsnmpfilter, 156
system addtelnetfilter, 156
system addudprelay, 157
system admin, 157
system authen, 157
system bootpserver, 158
system community, 158
system delhostmapping, 158
system deliproutingtable, 159
system delsnmpfilter, 160
system deltelnetfilter, 160
system deludprelay, 161
system history, 161
system list, 162
system log, 162
system moveiproutingtable, 163
system msg, 162
system name, 163
system onewandialup, 164
system passwd, 164
system securitytimer, 164
system snmpport, 165
system supporttrace, 165
system telnetport, 166
system wan2wanforwarding, 166
tcp stats, 145
vers, 145
voice, 296
voice l2clear, 289
voice l2stats, 289
Configuration commands

Router, 152
configuration examples
IKE aggressive mode, 131
IKE main mode, 128
PPP with IP and IPX, 59
RFC 1483 with IP and Bridging, 67
configuration files, backup/restore, 276
configuration information
Dual-Ethernet router, 42
FRF8 + IP, 41
PPP + bridging, 35
PPP + IP, 31
PPP + IPX, 33
RFC 1483 + bridging, 38
RFC 1483 + IP, 36
RFC 1483 + IPX, 37
RFC 1483MER + IP, 39
RFC 1490 + bridging, 38
RFC 1490 + IP, 31, 33, 35, 36
RFC 1490 + IPX, 37
RFC 1490MER + IP, 39
configuration tables
dual-Ethernet router +IP routing, 56
FRF8 + IP routing, 54
mixed network protocols, 55
PPP + bridging, 48, 49
PPP + IP routing, 46
PPP + IPX routing, 47
RFC 1483/RFC 1490 + bridging, 52
RFC 1483/RFC 1490 + IP routing, 50
RFC 1483/RFC1 490 + IPX routing, 51
RFC 1483MER/RFC 1490MER + IP routing, 53
connecting the PC to the console port, 313
contacting technical support, 292
conventions for commands, 135

## D

debugging commands, 293
DHCP
commands, 226
concepts, 81
configuration, 81
Diffie-Hellman group, 125
Diffie-Hellman key exchange, 121
DLCI display command, 150
dmt commands, 297
dmt mode command, 212
dod command, 293
dual-Ethernet router, 213

## E

embedded operations channel, 223
Encapsulated Security Payload, 119
encapsulation methods, IPSec, 120
encapsulation modes, IPSec, 119
encapsulation options, 24
encryption
Diffie-Hellman, 101
ESP protocol for IPSec, 120
PPP DES (RFC 1969), 100
EOC, 223
erase command, 144
error messages, 290
ESP IPSec protocol, 119
eth add command, 167
eth delete command, 168
eth ip bindroute command, 169
eth ip mgmt command, 176
Ethernet commands, 167, 213
Ethernet IP address assignment, 168
Ethernet subnets, 76
export restriction, 120

## F

file system commands, 146
filter br commands, 246
filtering parameters for IKE IPSec policies, 127
firewall, IP filtering, 103
flash memory
copying, 294
recovery procedures, 277
Frame commands, 149
frame relay debug commands, 297
Frame relay statistics command, 150
Frame stats command, 138
frame voice command, 289
FRF8, 41

## G

G_DMT mode setting, 297
G_LITE mode setting, 297

## H

history log, 282
host mapping, 95

## I

IAD, 18
IKE
    command formats, 248
    IPSec policy commands, 127
    IPSec proposal commands, 125
    peer commands, 124
    proposal commands, 125
    proposal exchange, 123
    protocol, 119
IKE configuration examples
    aggressive mode, 131
    main mode, 128
IKE flush, 248
IKE IPSec policies add, 248
IKE IPSec policies delete, 248
IKE IPSec policies disable, 248
IKE IPSec policies enable, 249
IKE IPSec policies list, 249
IKE IPSec policies set dest, 249
IKE IPSec policies set destport, 250
IKE IPSec policies set mode, 250
IKE IPSec policies set peer, 250
IKE IPSec policies set proposal, 251
IKE IPSec policies set protocol, 251
IKE IPSec policies set source, 251
IKE IPSec policies set sourceport, 252
IKE IPSec proposals add, 252
IKE IPSec proposals delete, 252
IKE IPSec proposals list, 252
IKE IPSec proposals set ahauth, 253
IKE IPSec proposals set espenc, 254
IKE IPSec proposals set ipcomp, 254
IKE IPSec proposals set lifedata, 254
IKE IPSec proposals set lifetime, 255
IKE peers add, 255
IKE peers delete, 255
IKE peers list, 255
IKE peers set address, 256
IKE peers set localid, 256
IKE peers set localidtype, 257
IKE peers set mode, 257
IKE peers set peerid, 257
IKE peers set peeridtype, 258
IKE peers set secret, 258
IKE proposals add, 258
IKE proposals delete, 259
IKE proposals list, 259
IKE proposals set dh_group, 259
IKE proposals set encryption, 260
IKE proposals set lifetime, 260
IKE proposals set message_auth, 260
IKE proposals set session_auth, 261
Integrated Access Device, 18
Internet Key Exchange protocol, 119
IP address assignment, 168
IP filter command
    Ethernet interface, 172
    WAN interface, 191
IP filtering, 103
IP filtering debug commands, 298
IP Firewall configuration, 79
IP routing
    configuration information (for dual-Ethernet router), 43
    configuration information (with FRF8), 41
    configuration information (with RFC 1483), 36
    configuration information (with RFC 1483MER), 39
    configuration information (with RFC 1490), 31, 33, 35, 36
    configuration information (with RFC 1490MER), 39
    configuration table (with FRF8), 54
    configuration table (with MAC Encapsulated Routing), 53
    configuration table (with RFC 1483/RFC 1490), 50
    configuration table (with RFC 1483MER/RFC 1490MER), 53
    configuration tables (with PPP), 46

test, 57
IP routing table, 77
defining, 154
deleting, 159
moving, 163
IP subnets, 76
IP virtual router support, 77
IP virtual routing, 169, 179, 186, 207
ipdebug command, 293
IPSec
command formats, 261
connection without IKE, 132
RFCs, 24
security, 119
IPSec add, 261
IPSec del, 261
IPSec disable, 261
IPSec enable, 262
IPSec flush, 262
IPSec list, 262
IPSec set authentication, 263
IPSec set authkey, 263
IPSec set compression, 263
IPSec set direction, 264
IPSec set enckey, 264
IPSec set encryption, 264
IPSec set gateway, 264
IPSec set ident, 265
IPSec set mode, 265
IPSec set service, 265
IPX routing
concepts, 309
configuration information (with RFC 1483), 37
configuration information (with RFC 1490), 37
configuration table (with PPP), 47
configuration table (with RFC 1483/RFC 1490), 51
test, 58

## K

kernel
upgrade from the LAN, 273
upgrade from the WAN line, 275
key add command, 99

## L

L2TP, 106
commands, 238
configurations, 109
over IPSec, 120
LED sequence, 281
LMI command, 150
Logical Ethernet interface, 168
logical Ethernet interfaces, 76
logical interface
defining, 167
login password
reset, 285

## M

MAC Encapsulated Routing, 39
management IP address
for a remote router, 203
for Ethernet interface, 176
management security, 97
minutes, 165
mlp debug command, 293
MTU command, 177
multiple Ethernet subnets, 76
multiple IP subnets, 76
multiple routing tables, 77

## N

NAT, 91
Network Address Translation, 91
classic NAT, 95
configuration, 91
IP filtering, 104
masquerading, 91
network information
example, 69
sample worksheets, 69
non-fatal errors, 273

## P

PAP, 20
PAP/CHAP, 20
password example, 66
password recovery, 279
password reset, 279

passwords for sample configuration, 66
PCM voice encoding, 18
peer commands, IKE, 123
ping command, 141, 283
policy commands, IKE, 123
port translation, 91
PPP Link Protocol, 31
proposal commands, IKE, 123
protocol standards, 23

## R

RAW-IMAGE special file name, 294
reboot command, 143
Release 4 feature list, 3
remote commands, 182
remote router database
    definition, 29
remote setMgmtIpAddr command, 203
replay detection, 120
restoring configuration files, 276
RFC 1483, 31, 36
RFC 1483MER, 39
RFC 1490, 31, 36
RFC 1490MER, 39
RFCs supported, 23
Router configuration commands, 152

## S

sample configuration, 67
    IKE configuration, 128
sample configurations
    dual-Ethernet router with IP, 73
    PPP with IP and IPX, 59
    RFC 1483 with IP and bridging, 67
SAs, 119
save command, 143
save dod, 144
saving configuration files, 276
saving the configuration, 143
SDSL commands, 222
SDSL debug commands, 295
sdsl eoc command, 223
security
    general information, 21
    IKE, 119
    IPSec, 119
Security Associations, 119
Setting a management address
    for a remote router, 203
setting a management address
    for Ethernet interface, 176
SNMP
    client validation, 97
    features, 267
software option keys, 99
software options
    encryption, 100
    IP filtering, 103
    keys, 99
    L2TP tunneling, 106
source routing, 77
standards conformance, 23
subnets, 76
support, contacting, 292
Symmetric Digital Subscriber Line, 222
system commands, 153
system files, 26
system level commands, 136
system messages, 290

## T

T1.413 mode setting, 297
TCP/IP Routing
    source and remote addresses, 37
TCP/IP routing
    control, 63, 70
    source and remote addresses, 37
technical support, contacting, 292
telephony services, 18
    trouble-shooting, 288
Telnet, 268
Telnet client validation, 97
TFTP, 268
TFTP server, 268
time-stamped messages, 290
transport mode, 119
troubleshooting
    bridging, 286
    console, 284
    factory configuration, 284
    hardware problems, 284
    history log, 282

IP routing, 286
IPX routing, 287
login password, 285
normal LED sequence, 282
PC connection, 285
power light off, 281
remote network access, 286
terminal window display, 284
using LEDs, 281
using ping, 283
Trouble-Shooting voice routing, 288
tunnel mode for IPSec, 119
tunneling, 106
Dial User, 107
L2TP, 107
LAC, 107
LNS, 107
tunneling configurations, 109

## U

unbind IP virtual route command
for a remote interface, 207
for an Ethernet interface, 179
upgrading the software kernel, 273

## V

Virtual Private Network security, 119
Virtual route binding, 169
virtual routing table, 77
adding, 154
deleting, 159
moving, 163
VoDSL router, 18
voice l2clear command, 289
voice l2stats command, 289
voice router debug commands, 296
voice routing, 18
trouble-shooting, 288
Voice, frame command, 150
VPI/VCI
find value, 288
VPN, 106
security, 119

## W

web GUI debug commands, 294

## Y

Y2K compliance, 271